LaTeX

$$\int \frac{1}{x^2} dx =$$

EL ARTE DE EDITAR EN LATEX

HEBER MQ

LaTeX

Heber MQ

2023

El arte de editar en LaTeX

Título de la obra: LaTeX

Autor: Heber Mamani Quispe

Edición: 2023

Tamaño de libro: 20x26cm

Páginas: 637

Idioma: Español

PEDIDOS Y COMENTARIOS

Whatsapp: +591 63152441

Correo: herbermqh@gmail.com

Facebook: https://www.facebook.com/hebermqh

Dedicatoria



Heber Mamani Quispe



Estefa Quispe Apaza

El presente libro dedico con mucho cariño a mi mamá **Estefa Quispe Apaza** por brindarme su apoyo y su comprensión cuando pasé escribiendo este libro en lugar de pasar tiempo con mi mamá.

Heber Mamani Quispe

Contenido general

Contenido

LaTeX

\LaTeX → PROCESADOR → LaTeX

Contents

CAPÍTULO 4 HERRAMIENTAS DE FORMATO

Comando	Produce
\tiny tiny	tiny
\scriptsize scriptsize	scriptsize
\footnotesize footnotesize	footnotesize
\small small	small
\normalsize normalsize	normalsize
\large large	large
\Large Large	Large
\huge huge	huge
\Huge Huge	Huge

CAPÍTULO 5 DISEÑO DE PÁGINA

CAPÍTULO 6 TABLAS

p	q	$p \to q$
0	0	1
0	1	1
1	0	0
1	1	1

FLOTADORES — CAPÍTULO 7

Figure 1: Descripción de la figura

MODO MATEMÁTICO — CAPÍTULO 8

$$E=\frac{a^2-1}{n^2+an}\left(\frac{1}{1-\frac{1}{n}}-1\right)\frac{a-an^3-n^4+n}{1-a^2}$$

$$E=\frac{-\cancel{(1-a^2)}}{n(n+a)}\left(\frac{1}{\frac{n-1}{n}}-1\right)\frac{a+n-n^3(a+n)}{\cancel{1-a^2}}$$

$$E=\frac{-1}{\cancel{n}\cancel{(n+a)}}\left(\frac{1-\frac{n-1}{n}}{\frac{n-1}{\cancel{n}}}\right)\cancel{(a+n)}\left(1-n^3\right)$$

$$E=-1\left(\frac{1}{n(n-1)}\right)(1-n)\left(1+n+n^2\right)$$

ÍNDICE Y BIBLIOGRAFÍA — CAPÍTULO 11

Referencia bibliográfica a [2], como también podemos citar a [1].

References

[1] M. Goossens; F, Mittelbach; A. Samarin. *The LaTeXCompanion*. Addison-Wesley. 1993.
[2] L. Lamport. *LaTeX*. Addison-Wesley. 1996.

PAQUETES Y CLASES — CAPÍTULO 12

.sty .cls

DOCUMENTCLASS — CAPÍTULO 13

TEX — CAPÍTULO 14

TEX

CAPÍTULO 15 LaTeX AVANZADO

LaTeX

CLASES ESTÁNDARES — CAPÍTULO 16

book
report
article

CLASES PERSONALIZADAS — CAPÍTULO 17

LATEX Y NORMAS APA 1

LaTeX y Normas APA: Una Guía Práctica para la Utilización de la Plantilla artapa7

Heber Mamani Quispe

El arte de editar en LaTeX

Notas del autor

Este trabajo académico se presenta como un complemento didáctico al libro de LaTeX - El arte de editar en LaTeX

INTRODUCCIÓN A LATEX

1.1 ¿Qué es LaTeX?

TeX

TeX es una lenguaje de programación al igual que los lenguajes de programación python, C y java, y reconoce comandos primitivos y no macros. Donald Knuth en su libro TeXbook denomina a TeX (lenguaje de programación) como virgen que tiene la capacidad de reconocer primitivas. Generalmente el término TeX es utilizado para referirse a plainTeX. El sistema TeX ha sido creado por Donald Knuth para crear documentos impresos. TeX es un lenguaje de composición tipográfica para edición de textos o artículos científicos que contienen ecuaciones o expresiones matemáticas. En la actualidad, en la mayoría de los casos, se utiliza TeX para edición de textos científicos.

LaTeX

PlainTeX (formato de TeX) es un conjunto de macros desarrollado por Donald Knuth. El lenguaje de programación TeX no reconoce macros; los macros se definen por formatos y paquetes.

LaTeXno es un procesador de textos como Microsoft Word, sino que es un conjunto de macros agregados a PlainTeX que ya están disponibles para su uso y ha sido aumentado estes macros por Leslie Lamport.

`\documentclass`

Para escribir un artículo o un libro científico se utiliza los macros o comandos ya definidos aunque en algunas ocasiones es necesario modificar o crear nuevos macros de LaTeX.

Para indicar que un documento LaTeX es un libro, se debe utilizar el comando o macro `\documentclass{book}` y para indicar que es un artículo, se debe utilizar `\documentclass{article}`. Una vez invocado este comando, las páginas y otros aspectos del documento LaTeX se formatean automáticamente de tal modo que sea un libro o un artículo.

Documento LaTeX

Un documento LaTeXestá conformado por texto ordinario al igual que en word (también se incluyen las expresiones matemáticas) y un conjunto de comandos (se la denomina comando a una palabra que está precedida por el caracter " \") y entornos.

Comparación de un documento LaTeX y un documento cualquiera

Un documento cualquiera (puede ser un libro científico) casi siempre consta de expresiones matemáticas y la parte literaria. Estas expresiones matemáticas en LaTeXse escriben dentro del modo matemático y para inicializar y finalizar el modo matemático se debe utilizar el signo dólar, `$a+b+c=12$`, y como también se puede utilizar un entorno para inicializar y finalizar el modo matemático. Existen comandos definidos solo para modo matemático y para modo texto.

Ejemplo 1.1

```
Ecuación: $c^2 = a^2 + b^2$
```

Ecuación: $c^2 = a^2 + b^2$

1.2 Instalación de distribución TeX

Distribución TeX

Como mencionamos en la sección anterior, LaTeXes un conjunto de macros definidos mediante macros TeXy existen 3 distribuciones de TeX: texlive, miktex y mactex. El texlive se puede instalar en Windows, Linux y Mac; el miktex es una distribución para Windows, y el mactex para Mac OS X. Todas las pruebas de este libro se realizaron con texlive 2022 en archlinux.

Editor de texto

Ahora surge la pregunta, ¿cuál es la mejor distribución de TeX? Esta pregunta suele generar una gran discusión, ya que generalmente cada distribución tiene sus ventajas y desventajas. Por lo tanto, no existe una mejor distribución debido a que no existe una distribución con desventajas para cada usuario. La mejor distribución dependería de cada usuario.

Instalación de distribución Texlive

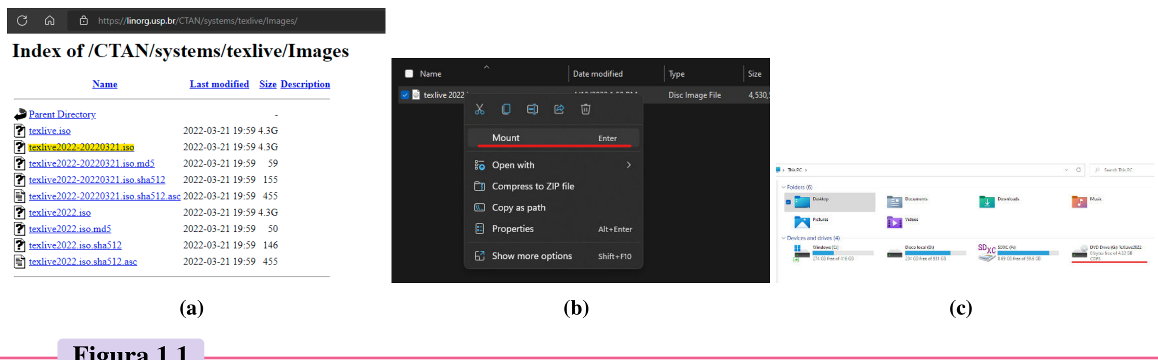
Windows

Instalación de Texlive en Windows

Pasos:

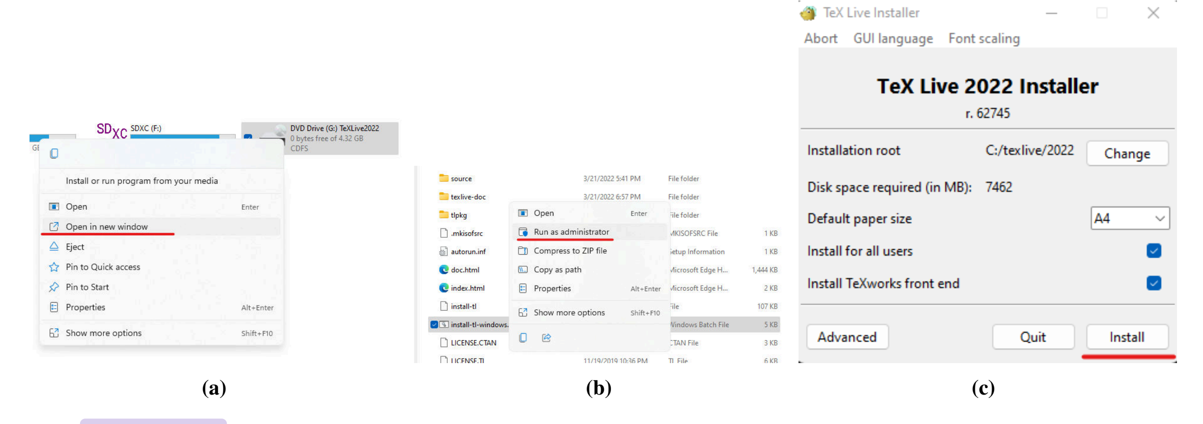
1. Descargar la imagen iso de texlive del siguiente enlace: `https://linorg.usp.br/CTAN/systems/texlive/Images/` (ver figura 1.1(a)).
2. Seleccionar la opción 2.
3. Esperar a que se complete la descarga.
4. Ir al directorio donde se ha descargado la imagen iso de texlive.
5. Montar el archivo ISO (ver figura 1.1(b)).
6. Ir al directorio donde se ha montado (ver figura 1.1(c)).
7. Entrar en la unidad montada (ver figura 1.2(a)).
8. Ejecutar como administrador el archivo `install-tl-windows.bat` (ver figura 1.2(b)).
9. Seleccionar la opción de Install”. También se puede realizar una instalación avanzada mediante la opción Advanced” (ver figura 1.2(c)).
10. Esperar a que se complete la instalación 1.3(a).
11. Una vez finalizada la instalación, simplemente debemos cerrar la ventana (ver figura 1.3).

Editores LaTeX



(a) (b) (c)

Figura 1.1

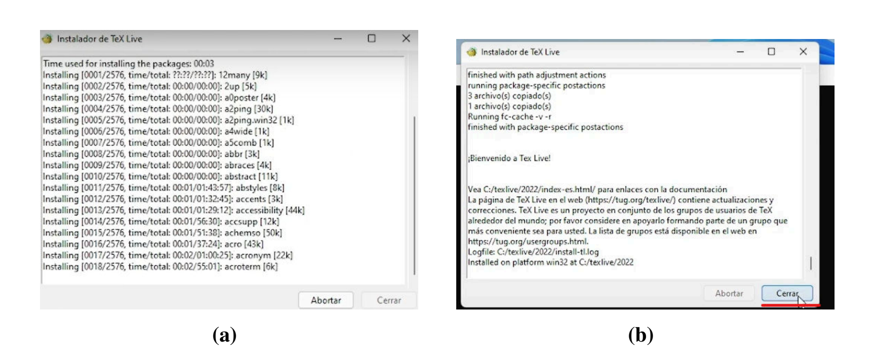


(a) (b) (c)

Figura 1.2

1.3 Editor LaTeX

Una vez ya instalada una distribución TEX, es necesario instalar algún editor LATEX. Existen varios editores LATEX, como texmaker, texstudio, Winshell, Kile y otros. Los editores mencionados ya vienen configurados; por lo tanto, buscan automáticamente la instalación de una distribución TEX. ¿Podemos configurar un editor de texto como un editor LATEX? Sí, es posible configurar un editor de texto como un editor LATEX, solo que en este caso debemos configurar para que reconozca la instalación de distribución TEX. Los editores de texto que se pueden configurar como un editor LATEXson bastantes, como Visual studio code, sublime, neovim y otros. Para escribir este libro se ha utilizado el editor neovim.



(a) (b)

Figura 1.3

1.4 Editor de expresiones matemáticas

Los editores "wysiwyg" de expresiones matemáticas surgieron debido a la imposibilidad de renderizar directamente las expresiones matemáticas en editores LaTeX. Para escribir este libro no se ha utilizado ningún editor wysiwyg de expresiones matemáticas.

El editor Tex equation editor es un editor wysiwyg de expresiones matemáticas y también existen otros editores wysiwyg (ver figura 1.4).

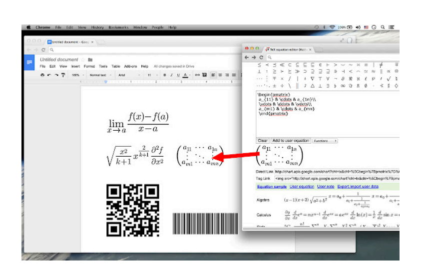


Figura 1.4

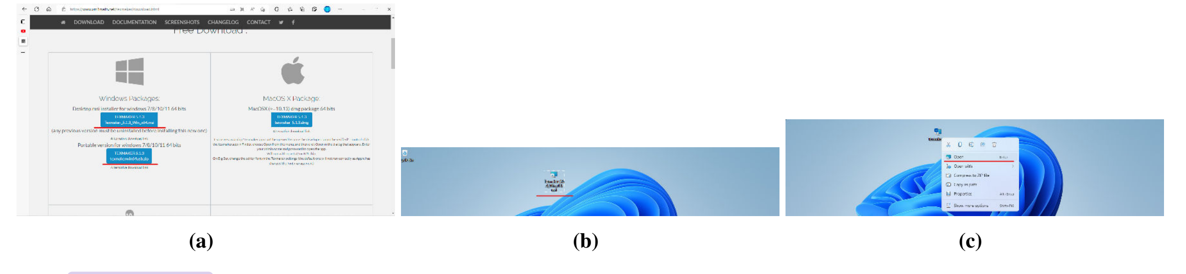
Tex equation editor

Instalación del editor Texmaker

Instalación del editor Texmaker

Pasos:

1. Descargar el archivo de instalación de Texmaker del siguiente enlace: `https://www.xm1math.net/texmaker/download.html`.
2. Seleccionar la opción de descargar dependiendo de la arquitectura del sistema (ver figura 1.5(a)).
3. Esperar a que se complete la descarga.
4. Ir al directorio en donde se ha descargado el archivo de instalación de Texmaker (en mi caso lo tengo descargado en el escritorio como se muestra en la figura 1.5(b)).
5. Ejecutar el archivo de instalación (ver figura 1.5(c)).



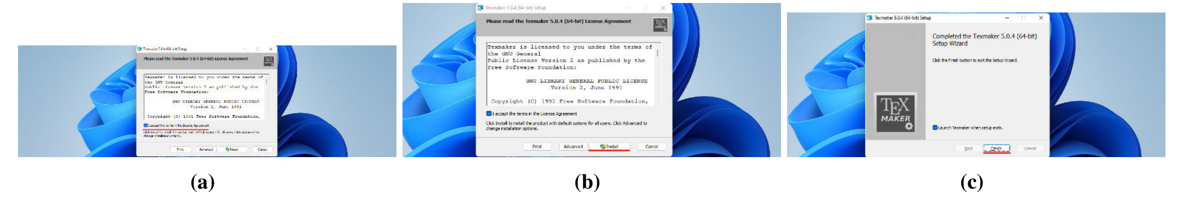
(a) (b) (c)

Figura 1.5

6. Aceptar los contratos y términos (ver figura 1.6(a)).
7. Seleccionar la opción de “Install” (ver figura 1.6(b)).

8 Esperar a que se complete la instalación.

9 Seleccionamos la opción de finalizar (ver figura 1.6(c)).



(a) (b) (c)

Figura 1.6

10 De esta forma hemos completado la instalación del editor texmaker (ver figura 1.7).

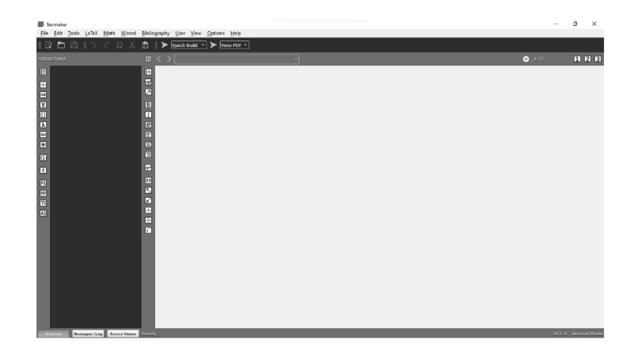


Figura 1.7

1.5 Editar y compilar un documento LATEX

Una vez ya instalada una distribución TEX y un editor LATEX, el siguiente paso es editar y compilar un documento LATEX. Para ello, supongamos que creamos el siguiente documento LATEX:

Primer documento LATEX

```
\documentclass{book}
\begin{document}
  Hola mundo.
\end{document}
```

Como resultado de la compilación en pdflatex, se obtiene un documento PDF con “Hola mundo”. Primero editamos un documento LATEX, luego lo compilamos y finalmente visualizamos los resultados (siempre y cuando no haya errores de compilación).

Compilación

Compilación de documento LATEX

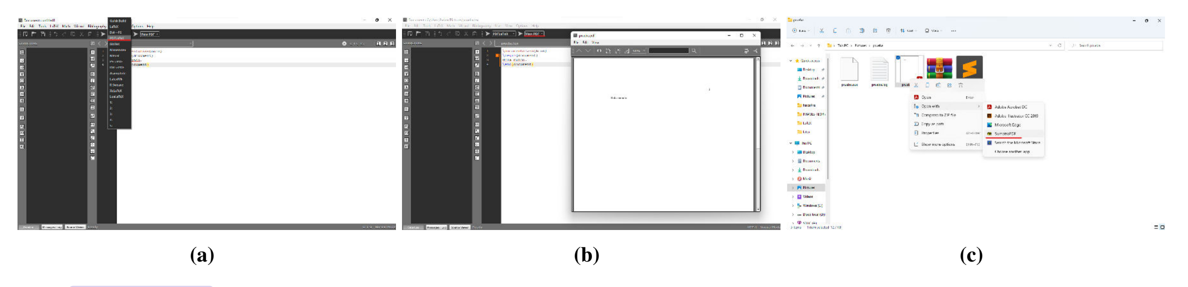
Al momento de tratar de compilar un documento LATEX, generalmente se muestran en los editores las opciones como “pdflatex, xelatex, lualatex y otras”, estos términos definen el motor de compilación a utilizar (en la mayoría de los casos se utiliza pdflatex) y estos motores no son de proceso multihilo por lo

motor de compilación

que suele tardar la compilación de un documento LaTeX. Xelatex tiene mejoras al momento de manejar fuentes y puede utilizar fuentes del sistema. Lualatex tiene la posibilidad de ejecutar scripts lua.

Compilar desde el editor Texmaker

Una vez ya creado y guardado un nuevo documento LaTeX, como siguiente paso es compilar este documento. Para ello, debemos seleccionar la opción de compilación en el editor Texmaker (ver figura 1.8(a)). En



(a) (b) (c)

Figura 1.8

este caso yo estoy seleccionando la opción de pdflatex, si quieres también puedes seleccionar una de las opciones restantes.

Resultado de la compilación en Texmaker

Abrir PDF en Texmaker

Abrir PDF con otro programa

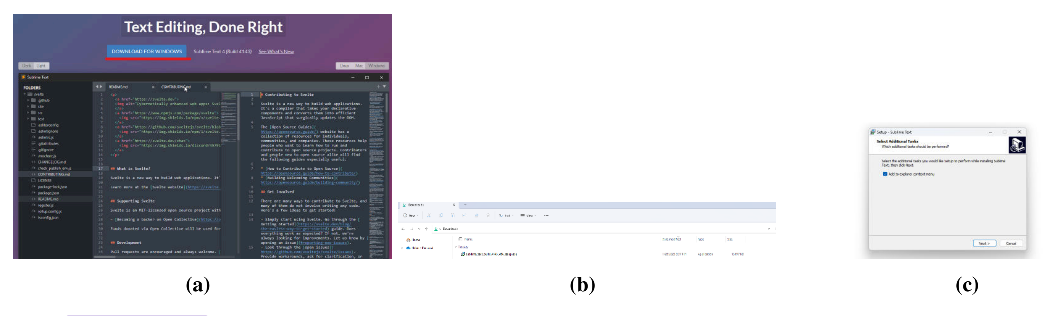
Una vez finalizada la compilación de un documento LaTeX, podemos ver el PDF obtenido a partir de la compilación del documento LaTeX. Para ver este PDF podemos seleccionar la opción de "View PDF" debido a que Texmaker viene incluido con un visualizador de PDF (ver figura 1.8(b)). También podemos utilizar un visualizador de PDF de nuestra elección, por lo que debemos ir al directorio en el que hemos guardado el documento LaTeXy abrir con un programa (de tu preferencia) el documento PDF generado (ver figura 1.8(c)).

1.6 Otros editores de texto

Sublime Text

Pasos:

1. Descargar el archivo de instalación de Sublime Text desde el siguiente enlace: `https://www.sublimetext.com/` (ver figura 1.9(a)).
2. Ejectuar el archivo de instalación (ver figura 1.9(b)).
3. Seleccionar "siguiente" (ver figura 1.9(c)).
4. Seleccionar "instalar" (ver figura 1.10(a)).
5. Esperar a que se complete la instalación.



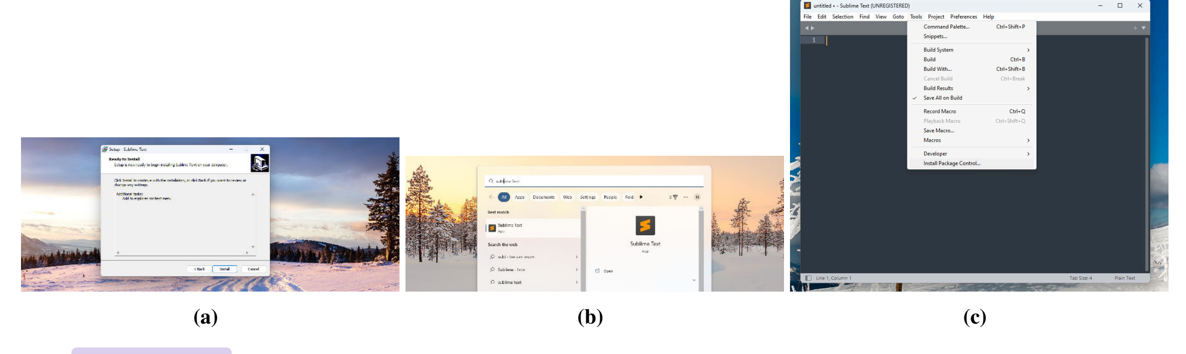
(a) (b) (c)

Figura 1.9

6 Seleccionar “finalizar”.

7 Abrir Sublime Text (ver figura 1.10(b)).

8 Instalamos control de paquetes de Sublime Text (ver figura 1.10(c)).



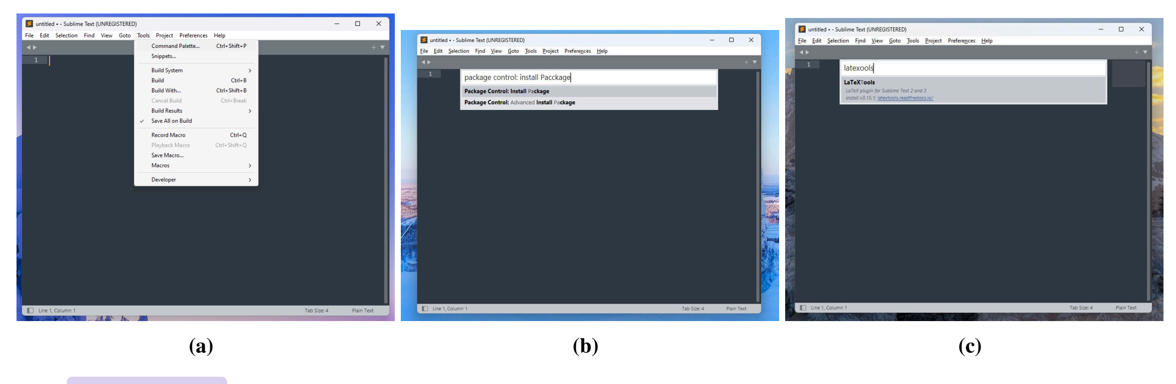
(a) (b) (c)

Figura 1.10

9 Nos vamos a paleta de comandos (ver figura 1.11(a)).

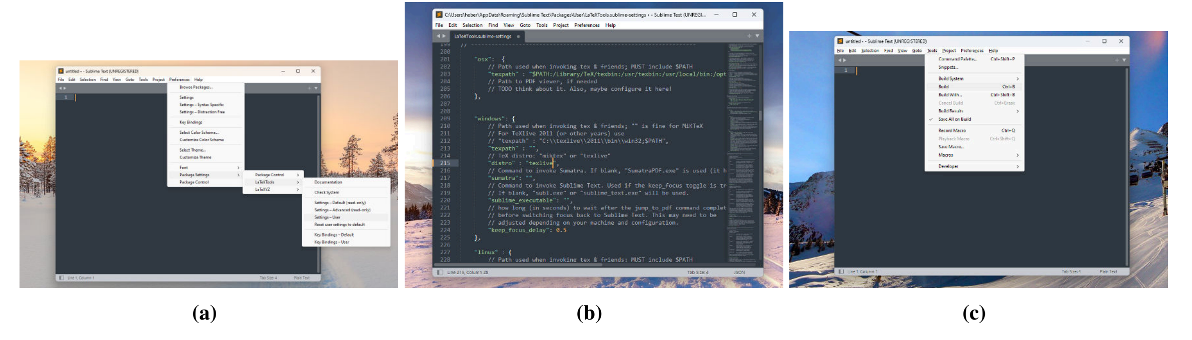
10 Seleccionamos la opción de “Package Control: Install Package” (ver figura 1.11(b)).

11 Buscamos y seleccionamos el paquete “LaTeXTools” (ver figura 1.11(c)).



(a) (b) (c)

Figura 1.11



(a) (b) (c)

Figura 1.12

12 Esperamos a que se complete la instalación.

13 De igual forma instalamos el paquete “LaTeXYZ”.

14 Se deseas puedes instalar más herramientas.

15 Nos vamos a las configuraciones de LaTeXTools (ver figura 1.12(a)).

16 Nos vamos a la línea 215 y cambiamos el valor de distro a texlive (ver figura 1.12(b)).

17 Guardamos el archivo y salimos.

18 Para compilar un documento LaTeX nos vamos a herramientas y luego a build (ver figura 1.12(c)).

Neovim

Para configurar neovim como un editor LaTeX se requiere de programas adicionales como “latexmk” y “xdotool.

Instalación de fuentes Hack Nerd Font:

1 Descargar el archivo de instalación de Hack Nerd Font desde el siguiente enlace: `https://www.nerdfonts.com/font-downloads` (ver figura 1.13(a))

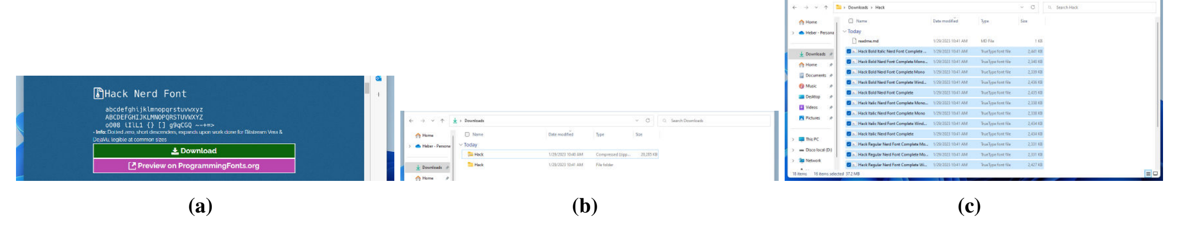
2 Nos vamos al directorio de descargas de nuestro sistema de archivos y extraemos el archivo Hack.zip (ver figura 1.13(b)).

3 Ingresamos al directorio Hack y copiamos el archivo Hack Nerd Font Complete.ttf (ver figura 1.13(c)).

4 Buscamos “Fonts” o “fuentes” en el menu inicio de nuestro sistema operativo (ver figura 1.14(a)).

5 Ingresamoa “Fonts”.

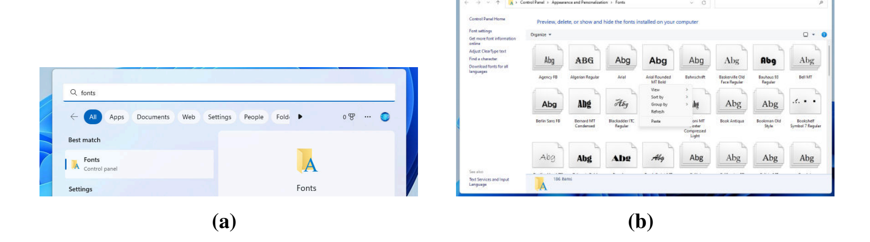
6 Pegamos los archivos copiados (ver figura 1.14(b)).



(a) (b) (c)

Figura 1.13

Figura 1.14

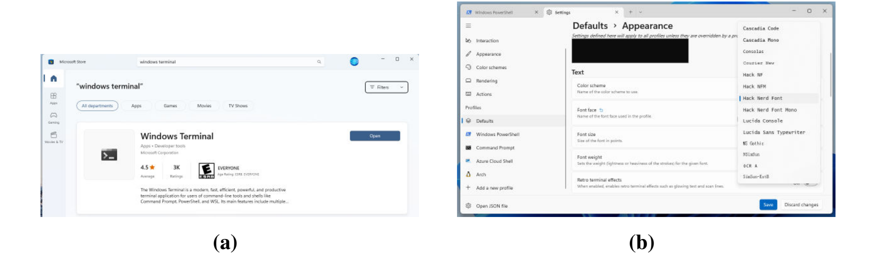


(a) (b)

Instalación de Windows terminal.

1. Buscamos en la tienda de Windows la aplicación de Windows terminal (ver figura 1.15(a)).
2. Seleccionamos la opción de instalar.

Figura 1.15



(a) (b)

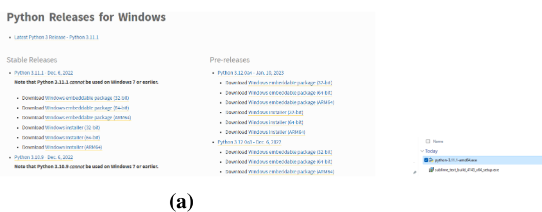
Las fuentes de terminal se debe cambiar al fuente Hack Nerd Font (ver figura 1.15(b)).

Instalación de python:

1. Descargamos el instalador de python desde el siguiente enlace: `https://www.python.org/downloads/windows/`.
2. Ejecutamos el instalador.
3. Seleccionamos la opción de “Customize installation”.
4. Seleccionamos siguiente.
5. Configuramos la instalación como se muestra en la figura.

Figura 1.16

Instalación de python


(a)

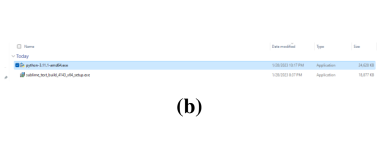
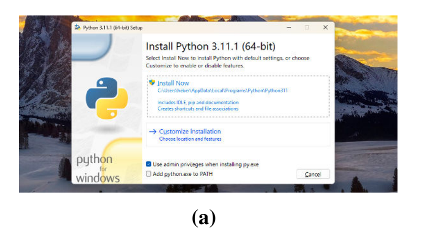
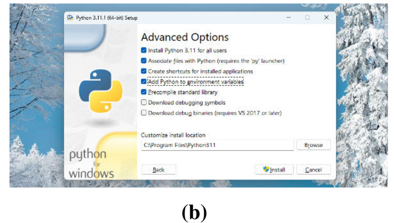

(b)

Figura 1.17

Instalación de python


(a)


(b)

6. Seleccionamos "Install".
7. Esperamos a que se complete la instalación.

Instalación de git:

1. Nos vamos a esta dirección `https://git-scm.com/downloads`.
2. Seleccionamos la opción Windows (ver figura 1.18(a)).

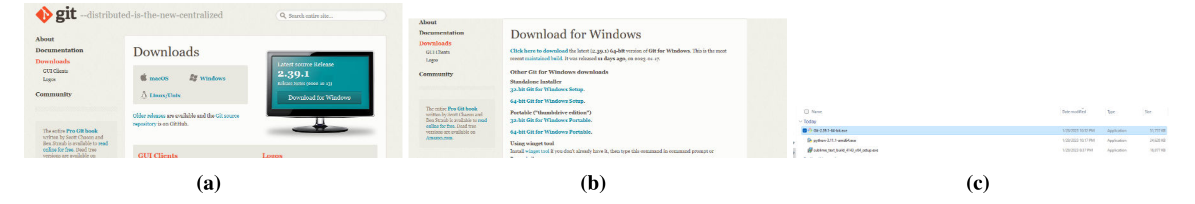
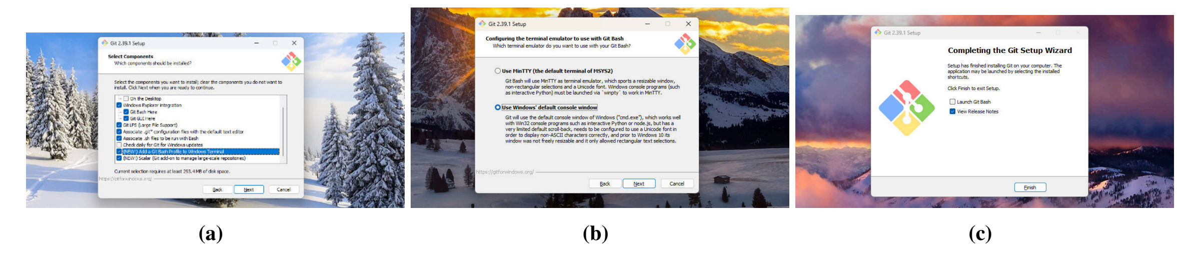

(a) (b) (c)

Figura 1.18

3. Seleccionamos la opción de "64-bit Git for Windows Setup" (ver figura 1.18(b)).
4. Ejecutamos el instalador (ver figura 1.18(c)).
5. Seleccionamos la opción de "Next".
6. Seleccionamos la opción de "Next".
7. Marcamos la opción de "Add a Git Bash Profile in Windows Terminal" (ver figura 1.19(a)).
8. Seleccionamos la opción de "Next".
9. Seleccionamos la opción de "Next".
10. Seleccionamos la opción de "Next".



(a) (b) (c)

Figura 1.19

11 Seleccionamos la opción de “Next”.

12 Seleccionamos la opción de “Next”.

13 Seleccionamos la opción de “Next”.

14 Seleccionamos la opción de “Next”.

15 Seleccionamos la opción de “Next”.

16 Seleccionamos la opción “Use Windows default console window” (ver figura 1.19(b).

17 Seleccionamos la opción de “Next”.

18 Seleccionamos la opción de “Next”.

19 Seleccionamos la opción de “Next”.

20 Seleccionamos la opción de “Next”.

21 Seleccionamos la opción de “Install”.

22 Finalizamos la instalación (ver figura 1.19(c)).

Instalación de gcc:

1 Descargamos el archivo rar del siguiente enlace: `https://drive.google.com/file/d/1kppslq5hZLx2A-u9LjAKr8EC` `view?usp=share_link`.

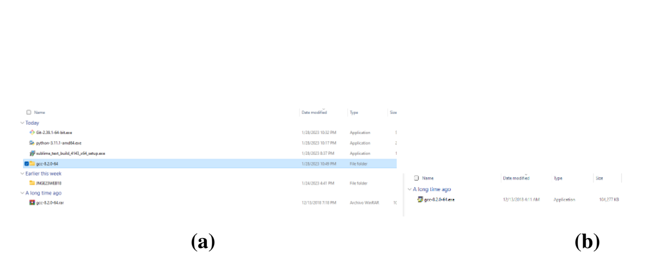
2 Extraemos el archivo (ver figura 1.20(a)).

3 Nos vamos a la carpeta de “gcc-8.2.0-64” (ver figura 1.20(b)).

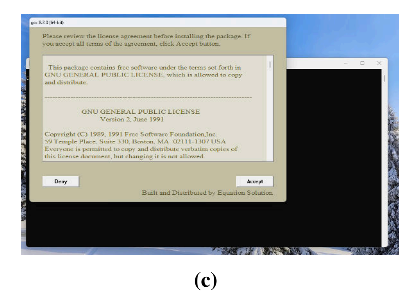
4 Ejecutamos el instalador.

5 Aceptamos los contratos de licencia (ver figura 1.20(c)).

6 Seleccionamos la opción “Install”.



(a) (b)



(c)

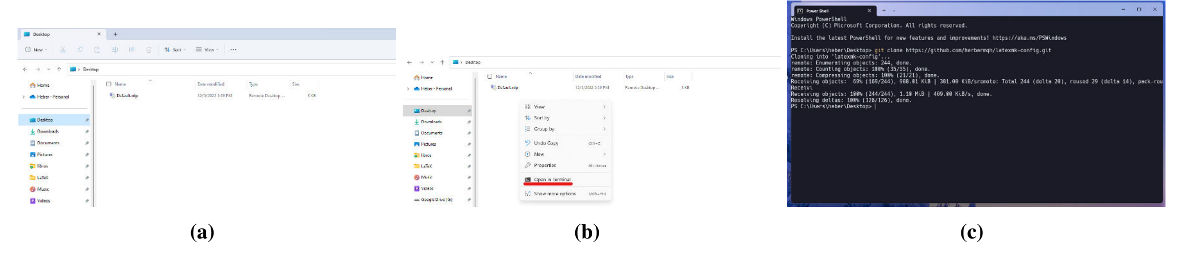
Figura 1.20

7 Esperamos a que se complete la instalación.

8 Seleccionamos la opción de "Finish".

Instalación de latexmk:

1 Nos vamos al directorio de "`C:\Users\Usuario\Desktop`" (ver figura 1.21(a)).



(a) (b) (c)

Figura 1.21

2 Abrimos terminal de Windows haciendo click derecho (ver figura 1.21(b))

3 Clonamos el repositorio `https://github.com/herbermqh/latexmk-config` pegando el siguiente comando en la terminal (ver figura 1.21(c)):

```
git clone https://github.com/herbermqh/latexmk-config.git
```

4 Cerramos la terminal.

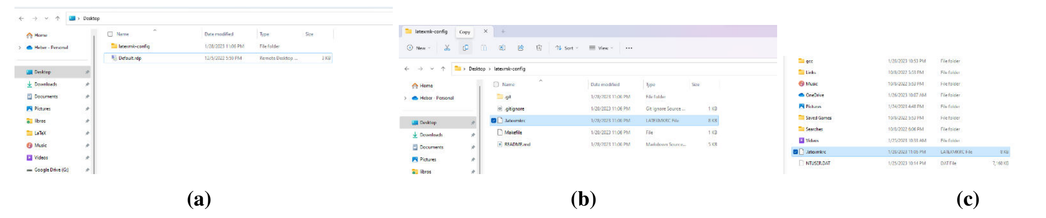
5 Nos vamos a la carpeta de "latexmk-config" (ver figura 1.22(a)).

6 Copiamos el archivo ".latexmkrc" (ver figura 1.22(b)).

7 Pegamos el archivo en el directorio "`C:\Users\Usuario`" (ver figura 1.22(c)).

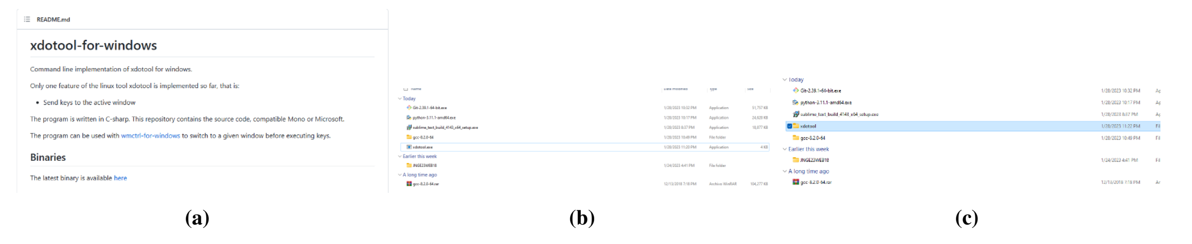
Instalación de xdotool:

1 Vamos a la siguiente dirección url `https://github.com/ebranlard/xdotool-for-windows`



(a) (b) (c)

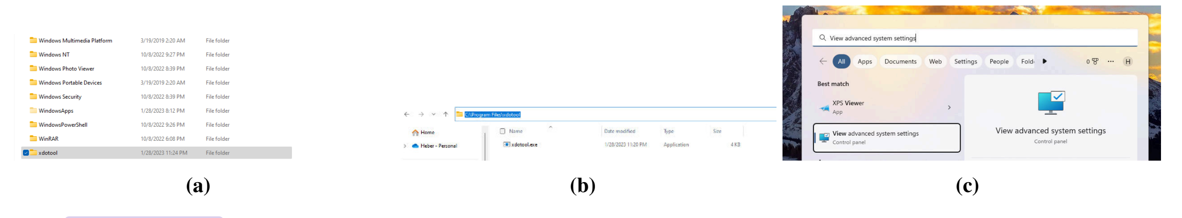
Figura 1.22



(a) (b) (c)

Figura 1.23

2. Descargamos haciendo click en la opción “here” (ver figura 1.23(a)).
3. Vamos a la carpeta de descargas (ver figura 1.23(b)).
4. Guardamos el ejecutable en una carpeta de nombre xdotool (ver figura 1.23(c)).
5. Copiamos la carpeta xdotool en el directorio “`C:\Program Files`” (también puede un directorio de su preferencia como se muestra en la figura 1.24(a)).

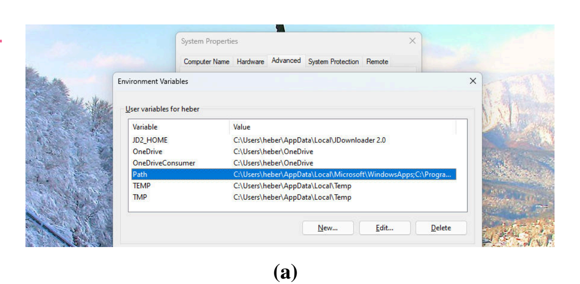


(a) (b) (c)

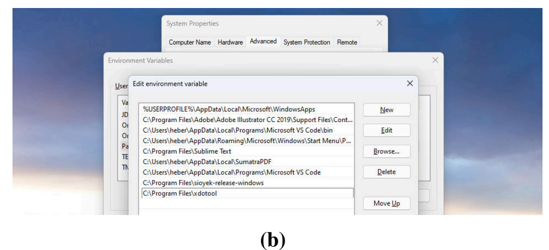
Figura 1.24

6. Ingresamos a la carpeta xdotool.
7. Copiamos la dirección de la carpeta (ver figura 1.24(b)).
8. Buscamos “ver configuración avanzada del sistema” en el buscador de Windows (ver figura 1.24(c)).
9. Seleccionamos la opción “Variables de entorno” .
10. En la ventana variables de usuario selecionamos la opción “Path” (ver figura 1.25(a)).
11. Seleccionamos la opción “Editar”.
12. Seleccionamos la opción “Nuevo”.

Figura 1.25



(a)



(b)

13 Pegamos la dirección de la carpeta xdotool (ver figura 1.25(b)).

14 Seleccionamos la opción “Aceptar”.

15 El mismo proceso realizamos para la ventana variables del sistema.

16 Salimos de la ventana variables de entorno haciendo click en la opción “Aceptar”.

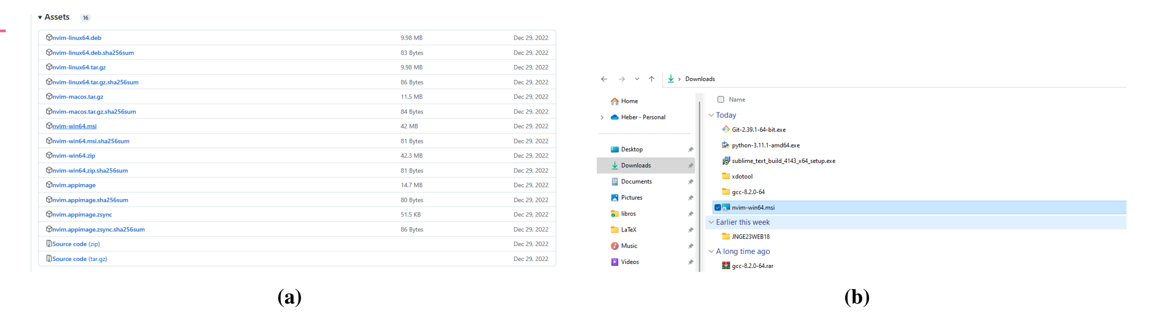
17 Salimos de la ventana propiedades del sistema haciendo click en la opción “Aceptar”.

Instalación de neovim:

1 Vamos a la siguiente dirección url `https://github.com/neovim/neovim/releases/tag/stable`

2 Decargamos el archivo “nvim-win64.msi” (ver figura 1.26(a)).

Figura 1.26



(a) (b)

3 Nos vamos a la carpeta de descargas (ver figura 1.26(b)).

4 Ejecutamos el instalador nvim-win64.msi.

5 Seleccionamos la opción “Next”.

6 Acepatamos los contratos de licencia.

7 Seleccionamos la opción “Next”.

8 Seleccionamos la opción “Next”.

9 Seleccionamos la opción “Install”.

10 Esperamos a que se complete la instalación.

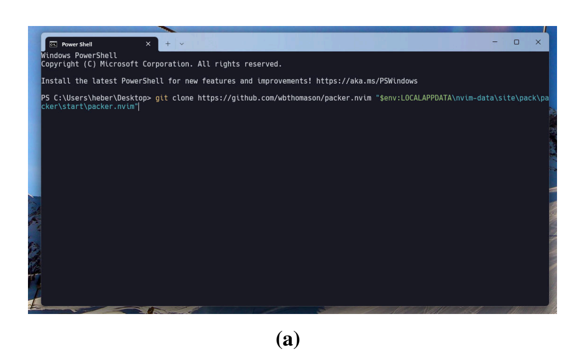
11 Seleccionamos la opción "Finish".

Instalación de gestor de paquetes packer:

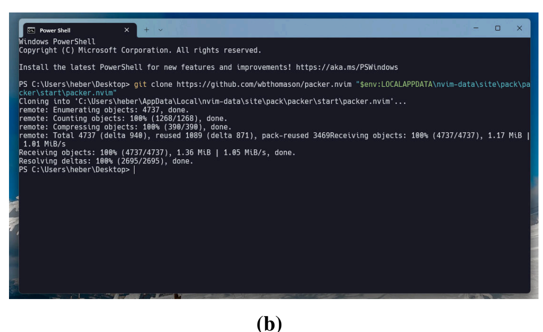
1 Abrimos terminal de Windows.

2 Clonamos el repositorio `https://github.com/wbthomason/packer.nvim` como se muestra en la figura 1.27(a):

Figura 1.27



(a)



(b)

3 Presionamos la tecla "Enter" (ver figura 1.27(b)).

4 Cerramos la terminal.

Instalación de pynvim:

1 Abrimos terminal de Windows.

2 Pegamos el siguiente comando en la terminal:

```
python -m pip install --user --upgrade pynvim
```

3 Presionamos la tecla "Enter".

4 Cerramos la terminal.

Instalación de nodejs:

1 Nos vamos a esta dirección url `https://nodejs.org/en/download/`

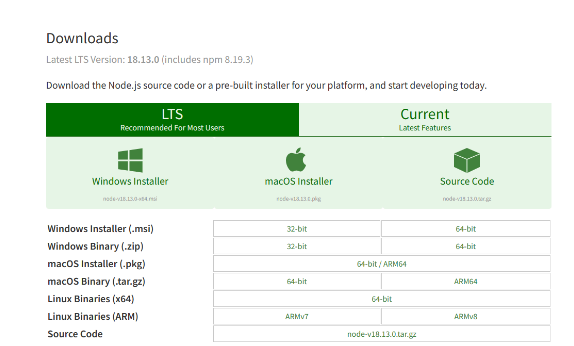
2 Hacemos click en la opción "Windows Installer" para descargar el instalador (ver figura 1.28).

3 Nos vamos a la carpeta de descargas.

4 Ejecutamos el instalador node.

5 Seleccionamos la opción "Next".

Figura 1.28



6 Acepatamos los contratos de licencia.

7 Seleccionamos la opción “Next”.

8 Seleccionamos la opción “Next”.

9 Seleccionamos la opción “Next”.

10 Seleccionamos la opción “Next”.

11 Seleccionamos la opción “Install”.

12 Esperamos a que se complete la instalación.

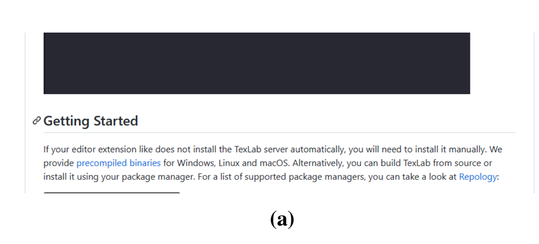
13 Seleccionamos la opción “Finish”.

Instalación de texlab:

1 Nos vamos a la siguiente dirección url `https://github.com/latex-lsp/texlab`

2 Hacemos click en “precompiled binaries” (ver figura 1.29(a)).

Figura 1.29



(a)



(b)

3 Hacemos click en la opción “`texlab-x86_64-windows.zip`” para descargar el archivo (ver figura 1.29(b)).

4 Nos vamos a la carpeta de descargas.

5 Descomprimimos el archivo `texlab-x86_64-windows.zip`.

6 Nos vamos a la carpeta `texlab-x86_64-windows`.

7 Copiamos el ejecutable texlab.exe.

8 Nos vamos al directorio “`C:\Program Files\xdotool`”.

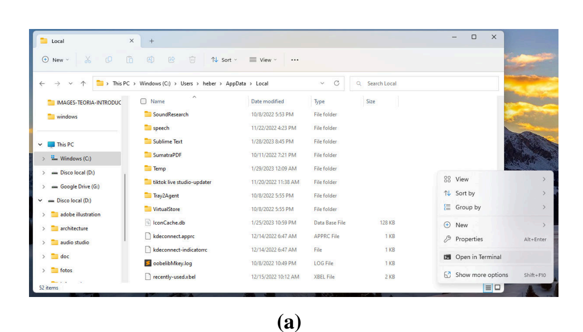
9 Pegamos el ejecutable texlab.exe.

Configuración de neovim:

1 Nos vamos al directorio “`C:\Users\Usuario\AppData\Local`”.

2 Click derecha y abrimos terminal de Windows (ver figura 1.30(a)).

Figura 1.30



(a)

(b)

3 Pegamos el siguiente comando en la terminal:

```
git clone https://github.com/herbermqh/hnvim.git nvim
```

Luego presionamos la tecla “Enter” (ver figura 1.30(b)).

4 Cerramos la terminal.

Iniciar en neovim:

1 Abrimos terminal de Windows.

2 Pegamos el siguiente comando en la terminal:

```
nvim
```

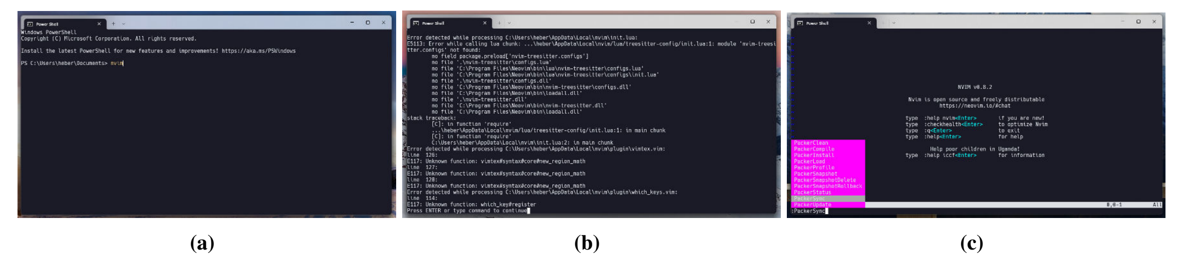
Como se muestra en la figura 1.31(a)

3 Presionamos la tecla “Enter” (ver figura 1.31(b)).

4 Presionamos la tecla “Enter”.

5 Sincronizamos los paquetes con el gestor de paquetes packer (ver figura 1.31(c)):

```
:PackerSync
```



(a) (b) (c)

Figura 1.31

6. Esperamos a que se sincronicen los paquetes.

7. Salimos de neovim con la combinación de teclas ":q!".

8. Volvemos a abrir neovim por medio de comando "nvim".

9. Presionamos la tecla "Enter" y ya tenemos neovim configurado como editor LaTeX (ver figura 1.32).



Figura 1.32

1.7 Preámbulo y cuerpo de un documento LaTeX

Preámbulo y cuerpo

Un documento LaTeX básicamente consta de 2 partes: un preámbulo y un cuerpo del documento LaTeX. En el preámbulo va el código que no se imprime en el documento pdf (si se compila en pdflatex), es decir, en el preámbulo del documento LaTeX van los códigos de configuración del documento LaTeX, como tamaños de letras, medidas de margen del artículo o libro y otros. En el cuerpo del documento LaTeX va todo código que se observa o se imprime en el documento PDF obtenido a partir de la compilación de un documento LaTeX.

Figura 1.33

```
\documentclass[10pt]{book}
\usepackage{amsmath}

\begin{document}
Teorema de Pitágoras: c² = a² + b²
\end{document}
```

preámbulo (líneas 1–2); cuerpo (líneas 4–6)

1.8 Paquetes LaTeX

Los paquetes LaTeX son archivos de extensión sty y contienen comandos y entornos definidos, es decir, estos archivos contienen una serie de códigos LaTeX y TeX. Los comandos y entornos definidos en un paquete se pueden utilizar en cualquier documento LaTeX, siempre y cuando el paquete se importe al documento LaTeX. Para importar un paquete se invoca el comando `\usepackage{nombre paquete}` en el que el argumento obligatorio es el nombre del paquete. Por ejemplo, supongamos que queremos importar el paquete `geometry` para redimensionar un documento LaTeX, entonces para realizar esta operación debemos tener un código como se muestra:

Paquete LaTeX es un conjunto de macros

Importar paquete

```
\documentclass{book}
\usepackage{geometry}
\begin{document}
  Hola mundo.
\end{document}
```

Dentro del archivo `geometry` están definidos ciertos comandos como `\newgeometry` y este comando se puede utilizar dentro de un documento LaTeX.

Los paquetes también se pueden ubicar en un directorio distinto, por lo que al invocar el comando `\usepackage` se debe especificar la ruta del paquete. Por ejemplo, supongamos que tenemos un paquete `geometry` en el directorio `/home/usuario/mypaquete.sty` y queremos importar este paquete en un documento LaTeX, entonces para realizar esta operación debemos tener un código como se muestra:

Importar un paquete desde un directorio

```
\documentclass{book}
\usepackage{/home/user/mypaquete}
\begin{document}
  Hola mundo.
\end{document}
```

MACROS LaTeX

El sistema TeX contiene una serie de comandos denominados primitivas y utilizando estos comandos podemos construir nuevos comandos o macros, así como también podemos construir nuevos paquetes y clases. LaTeX es un conjunto de macros que han sido construidos a partir de primitivas de TeX. En este capítulo nos centraremos en dar una breve introducción a la definición y manejo de nuevos macros TeX.

Sistema TeX y LaTeX

LaTeX no es un lenguaje de programación sino que es un texto sin formato y TeX es un lenguaje de expansión de macros. Algunos paquetes como tikz y pstricks utilizan TeX para algo que no ha sido diseñado para TeX (TeX ha sido diseñado para la composición tipográfica matemática y de texto). El paquete tikz llega a introducir el término "programación", pero el TeX sobre el que se construye no es un lenguaje de programación, lo que suele generar la confusión de que TeX es un lenguaje de programación.

2.1 Nombres de comandos y entornos

Los nombres de los comandos y entornos pueden llevar caracteres alfanuméricos, pero no pueden comenzar con un número. El nombre del comando "mycommand" es válido,

Nombres de comandos y entornos

```
\newcommand{\mycommand}{mi nuevo comando}
```

, y el nombre "3mycomando" no sería válido.

2.2 Tipos de comandos

Comandos simples

Los los comandos simples son de la forma:

1. `\bfseries`
2. `\sum`
3. `\tableofcontents`
4. `\newpage`

Comandos con uno más argumentos obligatorios

Los comandos con 1 o más argumentos son de la forma:

1. `\textbf{texto en negrita}`
2. `\frac{numerador}{denominador}`
3. `\includegraphics{imagen}`
4. `\chapter{título del capítulo}`

Comandos con argumetos opcionales

Los comandos con argumentos opcionales son de la forma:

1. `\documentclass[12pt]{article}`
2. `\includegraphics[width=0.5\textwidth]{imagen}`
3. `\chapter[texto]{nombre del capítulo}`
4. `\twocolumn[]`

Cuándo estemos utilizando el término comandos con argumentos se refiere tanto a comandos con argumentos obligatorios y opcionales.

Cuando definir nuevos comandos

2.3 Definir nuevos comandos

Los comandos generalmente se utilizan para tareas repetitivas. Por ejemplo,

Ejemplo 2.1

```
$x_1,x_2,x_3,\ldots,x_n$\\
$y_1,y_2,y_3,\ldots,y_n$\\
$z_1,z_2,z_3,\ldots,z_n$
```

$x_1,x_2,x_3,\ldots,x_n$
$y_1,y_2,y_3,\ldots,y_n$
$z_1,z_2,z_3,\ldots,z_n$

Las expresiones son repetidas, solo que cambian las letras x, y y z en las expresiones matemáticas.

`\newcommand`

La sintaxis para definir un nuevo comando es:

```
\newcommand{\NombreComando}[NumArg][ArgDefecto]{Definición del comando}
```

Argumento del comando `\newcommand`

El `\NombreComando` se debe reemplazar por el nombre que queremos asignar al comando. En la sección de `NumArg` debe ir el número de argumentos (el número de argumentos que admite el `\newcommand` es de 1 a 9 argumentos), cuando se utilizan dos o más argumentos el primer argumento se establece como opcional. En la sección de `ArgDefecto` va el valor por defecto para el argumento opcional. En la parte de la definición del comando se incluye todo lo relacionado con la tarea repetitiva o con la definición del comando en sí. Los argumentos que se utilizan en la definición del comando se representan mediante `#1,#2,#3,...,#9`.

Supongamos que queremos imprimir "$x_1,x_2,x_3,\ldots,x_n$" varias veces, por ello, tendríamos que definir un nuevo comando:

```
\newcommand{\expresion}{$x_1,x_2,x_3,\ldots,x_n$}
```

Para imprimir la expresión debemos llamar al comando `\expresion`.

Ejemplo 2.2

```
\newcommand{\expresion}{$x_1,x_2,x_3,\ldots,x_n$}
\expresion
```

$x_1,x_2,x_3,\ldots,x_n$

Ahora supongamos que queremos imprimir las expresiones

Ejemplo 2.3

```
$x_1,x_2,x_3,\ldots,x_n$\\
$y_1,y_2,y_3,\ldots,y_n$\\
$z_1,z_2,z_3,\ldots,z_n$
```

$x_1,x_2,x_3,\ldots,x_n$
$y_1,y_2,y_3,\ldots,y_n$
$z_1,z_2,z_3,\ldots,z_n$

por medio de la definición de un nuevo comando. Para ello debemos utilizar argumentos en el comando, por ejemplo:

Ejemplo 2.4

```
\newcommand{\expresion}[1]{$#1_1,#1_2,#1_3,\ldots,#1_n$}
\expresion{x}\\
\expresion{y}\\
\expresion{z}
```

$x_1,x_2,x_3,\ldots,x_n$
$y_1,y_2,y_3,\ldots,y_n$
$z_1,z_2,z_3,\ldots,z_n$

Argumentos obligatorios y opcionales

Los argumentos que van dentro de las llaves son argumentos obligatorios, mientras que los argumentos que van dentro de un corchete son argumentos opcionales.

Ahora supongamos que queremos imprimir las expresiones tal que el subíndice de x_n sea una letra cualquiera, para ello se debe utilizar dos argumentos:

Ejemplo 2.5

```
\newcommand{\expresion}[2]{$#1_1,#1_2,#1_3,\ldots,#1_#2$}
\expresion{x}{p}\\
\expresion{y}{q}\\
\expresion{z}{m}
```

$x_1,x_2,x_3,\ldots,x_p$
$y_1,y_2,y_3,\ldots,y_q$
$z_1,z_2,z_3,\ldots,z_m$

Si no queremos escribir el subíndice de x_n cada vez, pero en algún momento queremos cambiar el subíndice n, podemos utilizar argumentos opcionales. Por ejemplo:

Ejemplo 2.6

```
\newcommand{\expresion}[2][n]{$#2_1,#2_2,#2_3,\ldots,#2_#1$}
\expresion[p]{x}\\
\expresion[q]{y}\\
\expresion[m]{z}\\
\expresion{x}
```

$x_1,x_2,x_3,\ldots,x_p$
$y_1,y_2,y_3,\ldots,y_q$
$z_1,z_2,z_3,\ldots,z_m$
$x_1,x_2,x_3,\ldots,x_n$

Argumento opcional

Como podemos ver, el primer argumento #1 es opcional y el segundo argumento #2 es obligatorio. Esto ocurre cuando se incluye el segundo corchete.

2.4 Definir nuevos entornos

\newenvironment

La sintaxis para definir un nuevo entorno es:

```
\newenvironment{NombreEntorno}[NumArg][ArgDefecto]{Definición del entorno
↪ inicial}{Definición del entorno final}
```

Entornos más conocidos

Ejemplos de entornos:

1. Entorno de nombre `document`:

```
\begin{document}%inicio de entorno

\end{document}%fin de entorno
```

2. Entorno de nombre `enumerate`:

```
\begin{enumerate}%inicio de entorno

\end{enumerate}%fin de entorno
```

Un entorno se debe definir en preámbulo

Los entornos, al igual que los comandos, se deben definir en el preámbulo del documento LaTeX (esta no es una regla establecida, pero es una buena práctica definir entornos en el preámbulo), o bien pueden definirse en un paquete.

Argumentos del comando \newenvironment

Para definir nuevos entornos debemos utilizar el comando `\newenvironment` y seguir las instrucciones que se le indican en la sintaxis. En la sección de `NombreEntorno` va el nombre del entorno, en la sección de `NumArg` va el número de argumentos, al igual que en un comando, y finalmente en `ArgDefecto` va el argumento por defecto. En la sección "Definición del entorno inicial" van todos los órdenes que se ejecutarán antes de entrar al entorno, y en la sección "Definición del entorno final" van todos los órdenes que se ejecutarán al salir del entorno.

Supongamos que tenemos el siguiente fragmento de código:

Ejemplo 2.7

```
\begingroup
\bfseries
Hola mundo.
\endgroup
Fuera del grupo.
```

Hola mundo. Fuera del grupo.

Este fragmento de código se puede reemplazar definiendo un nuevo entorno:

Ejemplo 2.8

```
\newenvironment{myentorno}{\bfseries}{}
\begin{myentorno}
  Hola mundo.
\end{myentorno}
Fuera del grupo
```

Hola mundo. Fuera del grupo

El comando `\bfseries` solo tiene efecto dentro del entorno. El entorno "myentorno" se puede reutilizar en cualquier parte del documento.

Ejemplo 2.9

```
\newenvironment{myentorno}{\bfseries}{}
\begin{myentorno}
  Texto en negrita.
\end{myentorno}
```

Texto en negrita.

Los argumentos al momento de crear nuevos entornos se utilizan de manera similar a la de definir nuevos comandos; por ejemplo,

Ejemplo 2.10

```
\newenvironment{myentorno}[1]{\bfseries #1 \par}{}
\begin{myentorno}{órdenes antes del entorno}
  texto en negrita.
\end{myentorno}
```

órdenes antes del entorno
texto en negrita.

La diferencia entre el manejo de argumentos de los comandos `\newcommand` y `\newenvironment` es que el primer argumento de `\newenvironment` se puede establecer como opcional sin la necesidad de tener dos argumentos; por ejemplo,

Diferencia de argumentos

Ejemplo 2.11

```
\newenvironment{myentorno}[1][órdenes antes del
↪  entorno]{#1\par\bfseries}{}
\begin{myentorno}
  texto en negrita.
\end{myentorno}
```

órdenes antes del entorno
texto en negrita.

Ejemplo 2.12

```
\newenvironment{myentorno}[1][órdenes antes del
↪  entorno]{#1\par\bfseries}{}
\begin{myentorno}[argumento opcional]
  texto en negrita.
\end{myentorno}
```

argumento opcional
texto en negrita.

2.5 Redefinir comandos y entornos

Los comandos y entornos existentes se pueden redefinir.

\renewcommand

Sintaxis para ser redefinir un comando:

```
\renewcommand{\NombreComando}[NumArg][ArgDefecto]{Definición del comando}
```

\renewenvironment

Sintaxis para redefinir entorno:

```
\renewenvironment{NombreEntorno}[NumArg][ArgDefecto]{Definición del entorno
↪ inicial}{Definición del entorno final}
```

2.6 Contadores

Contadores

Como su nombre indica, un contador es el encargado de contar algo. En este caso, los contadores más conocidos pueden ser los contadores de páginas, los contadores de secciones, los contadores de tablas, figuras y otros. Cada contador tiene un nombre, valor y formato asociado.

Un contador se encarga de contar elementos

Formato de contadores

Código	Formato del contador
`\arabic{NombreContador}`	1,2,3,...
`\alph{NombreContador}`	$a,b,c,\ldots$
`\Alph{NombreContador}`	$A,B,C,\ldots$
`\roman{NombreContador}`	$i,ii,iii,\ldots$
`\Roman{NombreContador}`	$I,II,III,\ldots$
`\fnsymbol{NombreContador}`	$*,**,***,****$

Tabla 2.1 Formatos de contadores

Los contadores de formato `\alph` y `\Alph` no pueden superar el número de letras del abecedario, en este caso 27.

\roman

El contador de formato `\roman` mostrado en la tabla se obtiene cuando nuestro documento LaTeX está en inglés. Si se establece nuestro documento LaTeX a español con el paquete `babel` se obtendrán $I,II,III,\ldots$ en lugar de $i,ii,iii,\ldots$.

\fnsymbol

En el caso del último formato de los contadores se obtienen ,,,... cuando nuestro documento LaTeX es español, de lo contrario se obtendrán las marcas inglesas. Este formato de contador no puede superar el valor 6.

\the en los contadores

Para imprimir la representación de un contador debemos anteponer el prefijo `\the` al nombre del contador.

```
\the<NombreContador>
```

Por ejemplo,

```
\thechapter
```

Cuando se crea un nuevo contador en LaTeX por defecto se asigna el formato `\arabic` y si deseamos cambiar el formato debemos hacer un `\renewcommand*`

Redefinir formato de contador

El comando para imprimir el número de página (representación) es `\thepage` y si deseamos cambiar su formato debemos hacer un `\renewcommand*`. Por ejemplo,

Ejemplo 2.13

```
\documentclass{article}
\usepackage{geometry}
\geometry{left=5mm, right=5mm, top=5mm, bottom=5mm, paperwidth=12cm, paperheight=5cm}
\begin{document}
  Sin hacer un renewcommand*: \thepage

  \renewcommand*{\thepage}{\alph{page}}
  Realizando un renewcommand*: \thepage
\end{document}
```

Sin hacer un renewcommand*: 1
Realizando un renewcommand*: a

Manipulación de contadores

Operaciones con contadores

Para manipular los contadores existen las operaciones de incrementar un contador y asignar un valor o establecerlo.

Sintaxis para crear un nuevo contador:

`\newcounter`

```
\newcounter{<NombreContador>}[<ContadorExistente>]
```

El valor opcional `<ContadorExistente>` sirve para subordinar el nuevo contador `<NombreContador>` del `<ContadorExistente>`. El `<ContadorExistente>` ya debe estar creado. Por ejemplo, el contador de las secciones de un libro está subordinado del contador de los capítulos.

Sintaxis para asignar un valor al contador:

`\setcounter`

```
\setcounter{<NombreContador>}{<Valor>}
```

`\addtocounter`

Sintaxis para incrementar o decrementar el valor de un contador:

```
\addtocounter{<NombreContador>}{<Valor>}
```

El `valor` puede ser positivo o negativo.

`\value`

Sintaxis para recupera el valor de un contador:

```
\value{<NombreContador>}
```

Ejemplo 2.14

```
\documentclass{article}
\usepackage{geometry}
\usepackage{enumerate}
\geometry{left=5mm, right=5mm, top=5mm, bottom=5mm, paperwidth=12cm, paperheight=5cm}
\newcounter{nombreConservadorContador}
\begin{document}
  \begin{enumerate}
    \item Primer item
    \item Segundo item
    \setcounter{nombreConservadorContador}{\value{enumi}}
  \end{enumerate}
  Para recuperar el valor del contador nombreConservadorContador se debe establecer el contador enumi
  ↪  (contador de los items) al valor del contador nombreConservadorContador.
  \begin{enumerate}
    \setcounter{enumi}{\value{nombreConservadorContador}}
    \item Primer item
    \item Segundo item
  \end{enumerate}
\end{document}
```

1. Primer item
2. Segundo item

Para recuperar el valor del contador nombreConservadorContador se debe establecer el contador enumi (contador de los items) al valor del contador nombreConservadorContador.

3. Primer item
4. Segundo item

`\stepcounter`

El comando:

```
\stepcounter{<NombreContador>}
```

Incrementa el valor del `<NombreContador>` en uno y reinicia todos sus subordinados.

`\refstepcounter`

El comando:

```
\refstepcounter{<NombreContador>}
```

Realiza la misma operación que el comando `\stepcounter` solo que esta vez declara el valor del comando `\ref`.

Ahora vamos a crear un nuevo contador de líneas. Este contador se va a encargar de contar el número de líneas.

Ejemplo 2.15

```
\documentclass{article}
\usepackage{geometry}
\geometry{left=5mm, right=5mm, top=5mm, bottom=5mm, paperwidth=14cm, paperheight=5cm}
\newcounter{line}[section]
\newcommand{\lin}{\addtocounter{line}{1}\theline\quad}
\begin{document}
  \section{Primera sección}
  La representación del contandor line es: \theline\\
  \section{Prueba 1}
  \lin def sumaDosNumeros(a,b):\\
  \lin    return a+b
  \section{Prueba 2}
  \lin def sumaDosNumeros(a,b):\\
  \lin    return a+b
\end{document}
```

1 Primera sección

La representación del contandor line es: 0

2 Prueba 1

1 def sumaDosNumeros(a,b):
2 return a+b

3 Prueba 2

1 def sumaDosNumeros(a,b):
2 return a+b

2.7 Longitudes

Como su nombre indica, las longitudes tratan sobre longitudes; por ejemplo, la altura de una hoja de libro es una longitud. En LATEX existen las longitudes rígidas y las longitudes elásticas. Las longitudes rígidas no pueden adoptar otros valores al momento de compilar un documento LATEX, es decir, una longitud rígida nunca cambiará su valor durante la compilación. Las longitudes elásticas son todo lo contrario de las longitudes rígidas, es decir, pueden cambiar sus valores de acuerdo con los parámetros establecidos. Las longitudes elásticas se utilizan para que el documento LATEX se ordene de una forma más estética o se reacomode mejor.

Longitud rígida y elástica

Unidades de medida

Las unidades de medida utilizados en LATEX se muestra en la tabla 2.2.

Tabla 2.2

Unidad	Abreviatura	Equivalente a
Pulgada	in	25.4 mm o 72.27 puntos o 6.022 picas
Centímetro	cm	10 mm o 2.371 picas
Milímetro	mm	2.845 puntos
Punto	pt	0.351 mm
Pica	pc	12 puntos
Eme	em	Ancho de una M en la fuente vigente
Equi	ex	Altura de una x en la fuente vigente

Longitudes rígidas

Longitudes rígidas predeterminadas:

`\quad`	11.747 pt
`\thinspace`	1.958 pt
`\hoffset`	-28.45274 pt

Una longitud rígida se define de la siguiente manera `\hspace{<Valor><unidad>}`. Por ejemplo, `\hspace{2cm}`.

Longitudes elásticas

Las longitudes elásticas más conocidas son: `\bigskip`, `\medskip` y `\smallskip`.

Longitud elástica

Una longitud elástica se define como `\vspace{2cm plus 5mm minus 2mm}`. Este comando indica que se puede aumentar en 5 mm o reducir en 2 mm a partir de 2 cm. Esto no sucede con las longitudes rígidas.

`\hfill` y `\vfill`

Existen también longitudes elásticas de tipo `fil` y `fill`. Estas son longitudes elásticas infinitamente grandes. `fill` es más grande que `fil`. La longitud `fill` tiene un valor de `0 pt plus 1fill`.

Los comandos `\hfill` y `\vfill` son equivalentes a `\hspace{\fill}` y `\vspace{\fill}`, respectivamente.

Ejemplo 2.16

```
\documentclass{article}
\usepackage{geometry}
\geometry{left=5mm, right=5mm, top=5mm, bottom=5mm, paperwidth=14cm, paperheight=5cm}
\begin{document}
  | \hfill Heber \hfill |\\
  | \hspace{\fill} Heber \hspace{\fill} |
\end{document}
```

| Heber |
| Heber |

Definiendo y modificando longitudes

Sintaxis para definir nueva longitud:

\newlength

```
\newlength{\<NombreLongitud>}
```

Para obtener el valor de una longitud, se debe anteponer el comando \the al nombre de la longitud:

\the en longitudes

```
\the\<NombreLongitud>
```

Para establecer el valor de una longitud, se debe utilizar el comando \setlength.

\setlength

```
\setlength{\<NombreLongitud>}{<Valor>}
```

El valor <Valor> puede ser rígido, elástico o también puede ser otra longitud, como \textwidth. También existe otra forma de establecer el valor de una longitud, que consiste en igualar la variable de la longitud a un valor:

```
\<NombreLongitud>=<Valor>
```

o

Otra forma de establecer el valor de una longitud

```
\<NombreLongitud> <Valor>
```

para adicionar al valor de una longitud, se debe recurrir al comando \addtolength.

\addtolength

```
\addtolength{\<NombreLongitud>}{<Valor>}
```

El valor <valor> puede ser positivo o negativo.

Para establecer el valor de una longitud de un objeto, existen los siguientes comandos:

Longitud de objetos

```
\settowidth{\<NombreLongitud>}{<Objeto>}
\settoheight{\<NombreLongitud>}{<Objeto>}
\settodepth{\<NombreLongitud>}{<Objeto>}
```

Ejemplo 2.17

```
\newlength{\longtext}
\settowidth{\longtext}{aaaaa}
\begin{center}
  aaaaa\\
  aaaaa\hspace{\longtext}aaaaa\\
  aaaaa\hspace{\longtext}aaaaa\hspace{\longtext}aaaaa\\
  aaaaa\hspace{\longtext}aaaaa\\
```

```
    aaaaa
\end{center}
```

aaaaa

aaaaa aaaaa

aaaaa aaaaa aaaaa

aaaaa aaaaa

aaaaa

ESTRUCTURA DE DOCUMENTO

3.1 Estructura de documentos LaTeX

Un documento LaTeX está compuesto por dos secciones, preámbulo y un cuerpo del documento LaTeX.

Ejemplo 3.1

```
\documentclass{book}
\usepackage{amsmath,geometry}
\geometry{paperwidth=10cm,paperheight=3cm,margin=5mm}
\begin{document}
  Teorema de Pitágoras:
  \begin{align*}
    c^2 = a^2 + b^2
  \end{align*}
\end{document}
```

Teorema de Pitágoras:

$$c^2 = a^2 + b^2$$

3.2 Tipos de documentos estándares

Book

```
\documentclass[<opciones>]{book}
\title{<título del artículo>}
\author{<nombre del autor>}
\date{<...>}
```

```
\begin{document}
  \maketitle
  \chapter{<...>}
  \section{<...>}
  \subsection{<...>}
  \section{<...>}
  \subsection{<...>}
\end{document}
```

Article

```
\documentclass[<opciones>]{article}
\title{<título del artículo>}
\author{<nombre del autor>}
\date{<...>}
\begin{document}
  \maketitle
  \section{<...>}
  \subsection{<...>}
  \section{<...>}
  \subsection{<...>}
\end{document}
```

Report (reporte o informe)

Este clase estándar es lo mismo que la clase book, solo que están habilitados y deshabilitados algunas opciones por defecto. Se verá en detalle en la sección 3.4.

3.3 Paquetes recomendados

1. `babel`: para escribir en diferentes idiomas.
2. `inputenc`: para escribir caracteres especiales como caracteres acentuados.
3. `color`: este paquete nos permite usar colores en el documento LaTeX.
4. `amsmath`: para manejar expresiones matemáticas y ha sido creado por American Mathematical Society.
5. `amssymb`: para manejar símbolos matemáticos y ha sido creado por American Mathematical Society.
6. `amsthm`: para manejar teoremas, lemas, definiciones, etc.
7. `graphicx`: generalmente es utilizado para incluir figuras.

3.4 Opciones de clase book y report

1. Tamaño de letra: 10pt, 11pt y 12pt. Por defecto: 10pt.

2. Tamaño del papel: letterpaper (8.5 x 11 pulgadas, opción por defecto), legalpaper (8.5 x 14 pulgadas), executivepaper (7.25 x 10.5 pulgadas), a4paper (21 x 30 centímetros), a5paper (15 x 21 centímetros) y b5paper (18 x 25 centímetros).

3. Orientación del papel: portrait (vertical, opción por defecto) y landscape (horizontal).

4. Número de columnas: onecolumn (opción por defecto) y twocolumn.

5. Página separada para el título: titlepage y notitlepage. En los estilos book y report el título aparece en una página separada y mientras en article la opción por defecto es notitlepage.

6. Impresión en una o dos caras: oneside, twoside (opción por defecto).

7. Tipo de impresión: draft, final (opción por defecto). draft produce una caja negra en las líneas que se exceden del margen derecho (es solo advertencia)

8. Primera página de cada capítulo: openright y openany. El estilo book tiene por defecto openright y el estilo report tiene por defecto openany. La opción openright abre cada capítulo en el lado derecho del libro o en una página impar.

Documento LaTeX extenso

`\include` y `\input`

3.5 Modularización de un documento LaTeX

Un documento LaTeX extenso se puede dividir en archivos externos y luego, mediante el comando `\include` o `\input`, se pueden incluir esos archivos externos en el documento LaTeX principal o main.

Supongamos que tenemos el siguiente documento de LaTeX:

Ejemplo 3.2

```
\documentclass{book}
\usepackage{amsmath,geometry}
\geometry{paperwidth=15cm,paperheight=19cm,margin=5mm}
\begin{document}
  \chapter{INTRODUCCIÓN A \LaTeX}
  Contenido del primer capítulo.
  \chapter{MODO MATEMÁTICO}
  Contenido del segundo capítulo.
\end{document}
```

Chapter 1

INTRODUCCIÓN A LaTeX

Contenido del primer capítulo.

Chapter 2

MODO MATEMÁTICO

Contenido del segundo capítulo.

A medida que se vaya creando más capítulos, el documento LATEX se hará más extenso; por ello, surge la necesidad de separar cada capítulo en archivos externos (no siempre tienen que ser capítulos, también puede ser otro fragmento de código). Supongamos que tenemos creado el documento LATEX con el nombre `main` en una carpeta (ver figura 3.1). En la misma carpeta, se debe crear un nuevo archivo LATEX en el que

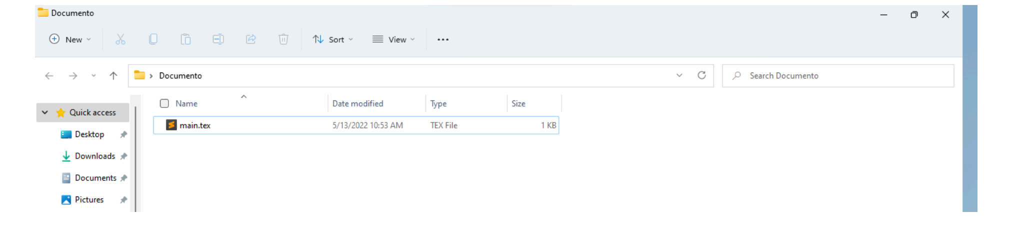


Figura 3.1

se incluirá un capítulo y luego posteriormente se debe incluir este archivo en el documento LATEX principal. Creamos dos archivos LATEX como se muestra en la figura 3.2. Ahora debemos incluir los archivos creados,

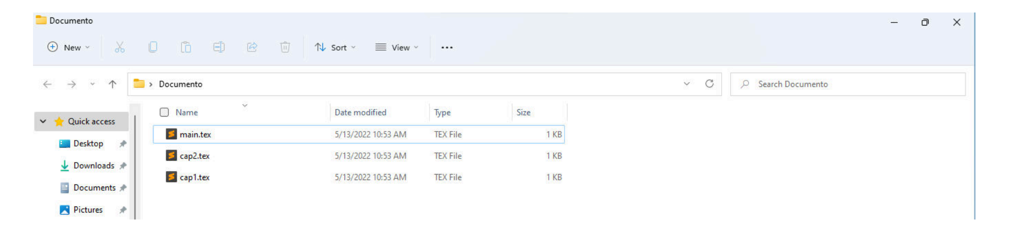


Figura 3.2

en este caso cap1 y cap2, en el documento LATEX principal:

```
\documentclass{book}
\usepackage{amsmath}
\begin{document}
  \include{cap1}
  \include{cap2}
\end{document}
```

Archivo ubicado en un directorio superior

También podemos incluir archivos `tex` ubicados en directorios superiores.

```
\input{../cap1.tex}
\input{../../cap2.tex}
```

Diferencias entre `\include` y `\input`:

1. `\input` es un macro de nivel más bajo que `\include`.
2. `\input` hace que se procese el archivo LATEX como si estuviera escrito en el documento LATEX principal, por lo que se puede invocar este comando en cualquier parte del documento LATEX.
3. `\include` hace que se procese el contenido del archivo y se ejecute el comando `\clearpage` antes y después, por lo que no se puede utilizar en cualquier parte del documento LATEX. Se recomienda utilizar `\include` para los capítulos de un libro debido a que tiene la ventaja de generar su propio archivo `.aux`, que luego es utilizado por el archivo `.aux` principal.
4. `\input` tiene la capacidad de anidar `\input`, pero `\include` no tiene esta capacidad.
5. `\include` está definido internamente mediante el comando `\input`.

3.6 Secciones de un documento LATEX

Clase de documento LATEX

Las clases de LATEXcomo un libro (book) o informe (article) ya tienen ciertos comandos y entornos definidos, además, estas tienen una cierta estructura definida de acuerdo con la clase. La estructura de un documento LATEXcomo los capítulos, las secciones, el apéndice, la bibliografía y otros, se pueden redefinir. Por defecto, vienen definidos los mencionados de una forma estructural; por lo tanto, es necesario conocer estas estructuras para redefinirlas.

Comandos de estructura de documento LATEX:

1. El comando `\maketitle` genera el título del documento LATEX.
2. El comando `\chapter` genera un capítulo.
3. El comando `\section` genera una sección.
4. El comando `\subsection` genera una subsección.
5. El comando `\subsubsection` genera una subsubsección.

6 El comando `\appendix` genera un apéndice.

`\frontmatter`, `\mainmatter` y `\backmatter`

7 El comando `\bibliography` genera una bibliografía.

Un libro generalmente está dividido en 3 partes. En la primera parte van la portada, el índice y otros; en la segunda parte van los capítulos, y en la última parte suele tener el apéndice y la bibliografía. Para indicar al LATEX que se trata de la primera parte, se debe utilizar el comando `\frontmatter`; para la segunda parte, `\mainmatter`; y para la parte final, `\backmatter`.

Ejemplo 3.3

```
\documentclass{book}
\usepackage{amsmath,geometry}
\geometry{paperwidth=15cm,paperheight=19cm,margin=5mm}
\title{Título}
\begin{document}
  \frontmatter
  \maketitle
  \tableofcontents
  \mainmatter
  \chapter{INTRODUCCIÓN A \LaTeX}
  Contenido del primer capítulo.
  \chapter{MODO MATEMÁTICO}
  Contenido del segundo capítulo.
  \backmatter
  \appendix
  \bibliography{bib}
\end{document}
```

Título

June 4, 2023

Contents

Chapter 1

INTRODUCCIÓN A LaTeX

Contenido del primer capítulo.

Chapter 2

MODO MATEMÁTICO

Contenido del segundo capítulo.

Nivel de `\part` en un artículo

El nivel del comando `\part` para un artículo es 0.

Los comandos de la forma `\section*` se invocan de manera automática o se llaman internamente cuando se invocan los comandos como `\tableofcontents`, `\listoftables` y otros; no siempre pueden ser comandos, también pueden ser entornos como `thebibliography`.

Comando section

`\section`

`\section *`

La sintaxis del comando `\section` es:

```
\section*{<title>}
\section[<toc-entry>]{<title>}
```

El primer comando no genera las enumeraciones de la página y además no entra en la tabla de contenidos. El segundo comando genera las enumeraciones en las páginas y entra en la tabla de contenidos. El argumento opcional `<toc-entry>` es el texto que se genera en la tabla de contenido y en el encabezado.

Ejemplo 3.4

```
\documentclass{article}
\usepackage{amsmath,geometry}
```

```
\geometry{paperwidth=15cm,paperheight=10cm,margin=5mm}
\begin{document}
  \tableofcontents
  \section*{INTRODUCCIÓN A \LaTeX}
  Contenido del primer capítulo.
  \section{MODO MATEMÁTICO}
  Contenido del segundo capítulo.
  \section[Tablas LaTeX]{TABLAS}
\end{document}
```

Contents

INTRODUCCIÓN A LaTeX

Contenido del primer capítulo.

1 MODO MATEMÁTICO

Contenido del segundo capítulo.

2 TABLAS

3.7 Enumeraciones

Supongamos que tenemos el siguiente documento LaTeX:

Ejemplo 3.5

```
\documentclass{book}
\usepackage[paperwidth=15cm, paperheight=12cm, margin=5mm]{geometry}
\begin{document}
  \chapter{LA RECTA}
  \section{Ecuación de la recta}
  \subsection{Forma punto pendiente de la recta}
  \section{Distancia entre entre dos rectas}
  \subsection{Subseción de prueba}
\end{document}
```

Chapter 1

LA RECTA

1.1 Ecuación de la recta

1.1.1 Forma punto pendiente de la recta

1.2 Distancia entre entre dos rectas

1.2.1 Subseción de prueba

En el ejemplo podemos apreciar que cada capítulo, cada sección y subsección están enumerados. Estas enumeraciones se pueden manipular, pero no se recomienda. También podemos dar formato a estas enumeraciones.

Numeración de secciones

Las numeraciones expuestas están definidas de la siguiente forma:

Contadores de secciones

```
\newcounter{part}
\newcounter{chapter}
\newcounter{section}[chapter]
\newcounter{subsection}[section]
\newcounter{subsubsection}[subsection]
\newcounter{paragraph}[subsubsection]
\newcounter{subparagraph}[paragraph]
```

El contador `subsection` se resetea cada vez que se detecte otra sección. Se interpreta de manera análoga para los demás contadores.

Para resetear un contador se debe utilizar el comando `@addtoreset`.

```
\makeatletter
\@addtoreset{chapter}{part}
\makeatother
```

El contador chapter se reseteará cada vez que se encuentre una nueva parte.

Para cada contador de nombres chapter, part, section y subsection existe su forma de representación o formateado. El formato del contador se puede imprimir anteponiendo el prefijo `\the`.

`\the` en secciones

```
\thepart
\thechapter
```

```
\thesection
\thesubsection
\thesubsubsection
\theparagraph
\thesubparagraph
```

Ejemplo 3.6

```
\documentclass{book}
\usepackage[paperwidth=15cm, paperheight=12cm, margin=5mm]{geometry}
\begin{document}
  \chapter{LA RECTA}
  \section{Ecuación de la recta}
  \subsection{Forma punto pendiente de la recta}
  El nombre el contador subsection es: \thesubsection
  \section{Distancia entre entre dos rectas}
  \subsection{Subseción de prueba}
\end{document}
```

Chapter 1

LA RECTA

1.1 Ecuación de la recta

1.1.1 Forma punto pendiente de la recta

El nombre el contador subsection es: 1.1.1

1.2 Distancia entre entre dos rectas

1.2.1 Subseción de prueba

La forma de representación o formateado de los contadores se puede realizar de la siguiente forma:

Formateo de secciones

```
\renewcommand\thepart{\arabic{part}}
\renewcommand\thechapter{\arabic{chapter}}
\renewcommand\thesection{\thechapter.\arabic{section}}
\renewcommand\thesubsection{\thesection.\arabic{subsection}}
```

También puede ser de la siguiente forma:

```
\renewcommand\thepart{\@Roman\c@part}
\renewcommand\thesection{\@arabic\c@section}
```

```
\renewcommand\thesubsection{%
  \thesection.\@arabic\c@subsection%
}
\renewcommand\thesubsubsection{%
  \thesubsection.\@arabic\c@subsubsection%
}
\renewcommand\theparagraph{%
  \thesubsubsection.\@arabic\c@paragraph%
}
\renewcommand\thesubparagraph{%
  \theparagraph.\@arabic\c@subparagraph%
}
```

Modificando forma de representación del contador `section`:

Ejemplo 3.7

```
\documentclass{book}
\usepackage[paperwidth=15cm, paperheight=12cm, margin=5mm]{geometry}
\renewcommand\thepart{\arabic{part}}
\renewcommand\thechapter{\arabic{chapter}}
\renewcommand\thesection{\Alph{section}}
\renewcommand\thesubsection{\thesection.\arabic{subsection}}
\begin{document}
  \tableofcontents
  \chapter{LA RECTA}
  \section{Ecuación de la recta}
  \subsection{Forma punto pendiente de la recta}
  El nombre el contador subsection es: \thesubsection
  \section{Distancia entre entre dos rectas}
  \subsection{Subseción de prueba}
\end{document}
```

Contents

Chapter 1

LA RECTA

A Ecuación de la recta

A.1 Forma punto pendiente de la recta

El nombre el contador subsection es: A.1

B Distancia entre entre dos rectas

B.1 Subseción de prueba

Cuando se ha modificado el nombre del contador `section`, entonces también se han modificado la representación del contador `subsection`. Esto se debe a que el comando `\thesubsection` está afectado por el comando `\thesection`.

Efecto al modificar la representación de un contador

3.8 Formato de chapter, section y subsection

Para cambiar los diseños o formatos de los títulos de chapter, section, subsection y otros, LaTeX proporciona el comando `\@startsection`. Sintaxis del comando `@startsection`:

`\@startsection`

```
\@startsection{name}{level}{indent}{beforeskip}{afterskip}{style}
```

name es el nombre de la sección, por ejemplo, “section”.

level es el nivel de la sección, por ejemplo, capítulo = 0, sección = 1, etc.

indent es la sangría de la cabecera desde el margen izquierdo.

beforeskip es la cantidad de espacio que se deja encima de la cabecera. Si es negativo, se suprime la sangría del párrafo para el texto que sigue a la cabecera.

afterskip es si es positivo, se deja espacio debajo de la cabecera, de lo contrario es el valor negativo del espacio que se deja a la derecha de la cabecera en una línea.

style son comandos para establecer el estilo. El último comando en este argumento puede ser un comando como `\MakeUppercase` o `\fbox` que toma un argumento. La cabecera de la sección se proporcionará como argumento para este comando.

La función `@startsection` tiene una estructura de control de flujo que se utiliza para controlar el formato y la posición de la sección en el documento.

Ejemplo 3.8

```
\documentclass{article}
\usepackage[paperwidth=15cm, paperheight=7cm, margin=5mm]{geometry}
\usepackage{lipsum}
\makeatletter
\renewcommand\section{%
  \@startsection{section}% name
    {1}% level
    {0pt}% indent
    {-3.5ex plus -1ex minus -.2ex}% beforeskip
    {2.3ex plus.2ex}% afterskip
    {\centering\normalfont\Large\bfseries}% style
  }
\makeatother
\begin{document}
  \section{Prueba de sección}
  \lipsum[1]
\end{document}
```

1 Prueba de sección

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Ejemplo 3.9

```
\documentclass{article}
\usepackage[paperwidth=15cm, paperheight=17cm, margin=15mm]{geometry}
\usepackage{lipsum}
\newcounter{mysection}% counter for numbering
\makeatletter
\renewcommand{\themysection}%             how to display
  {\@arabic\c@mysection}%  numbering
\newcommand{\mysection}{%
  \@startsection{mysection}%
    {1}%
    {0pt}%
    {-3.5ex plus -1ex minus -.2ex}% beforeskip
    {2.3ex plus.2ex}% afterskip
    {\normalfont\Large\bfseries}
}
\newcommand*\l@mysection{\@dottedtocline{1}{10em}{5em}}% for toc
\newcommand{\mysectionmark}[1]{}% for page headers
\makeatother
\begin{document}
  \tableofcontents
  \section{Prueba de sección}
  \lipsum[1]
  \mysection{Primera prueba de mysection}
  \lipsum[1]
  \mysection{Segunda prueba de mysection}
  \lipsum[1]
\end{document}
```

Contents

1 Prueba de sección

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

1 Primera prueba de mysection

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean

1

faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

2 Segunda prueba de mysection

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

2

3.9 Paquete fncychap

fncychap

Sintaxis de uso de este paquete:

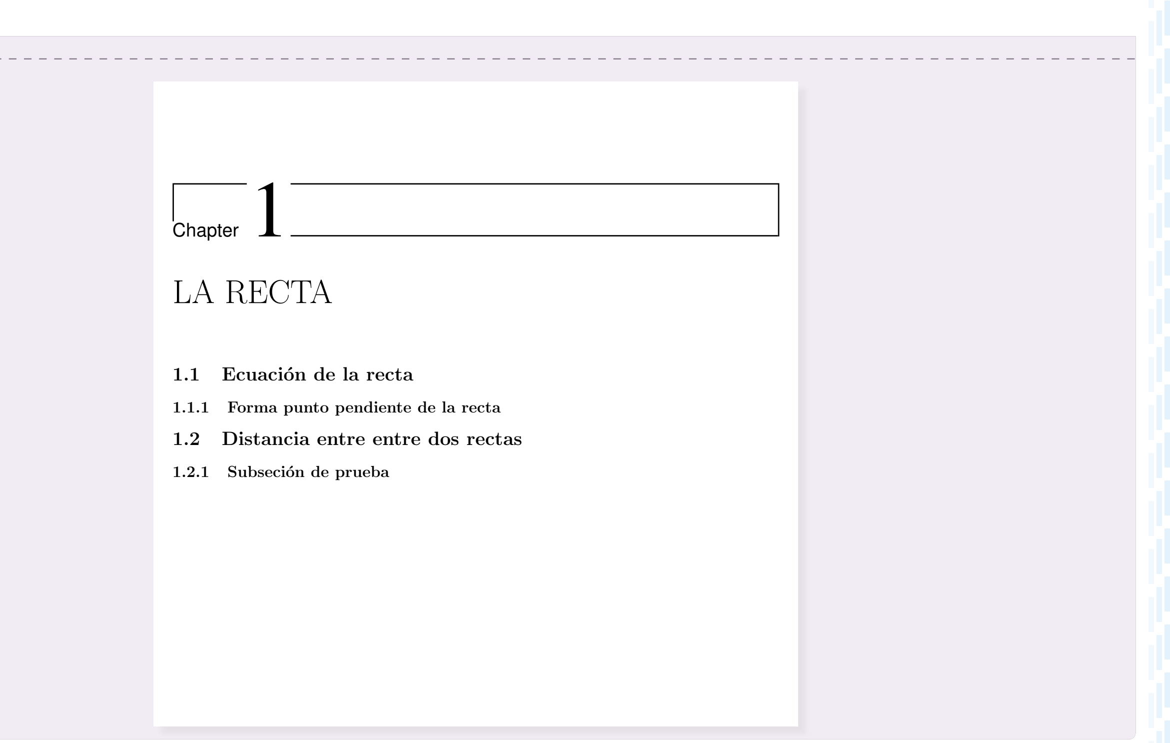
```
\usepackage[style]{fncychap}
```

Los estilos de capítulos predefinidos son: Sonny, Lenny, Glenn, Conny, Rejne y Bjarne.

Ejemplo de uso de este paquete:

Ejemplo 3.10

```
\documentclass{book}
\usepackage[paperwidth=17cm, paperheight=17cm, margin=5mm]{geometry}
\usepackage[Lenny]{fncychap}
\begin{document}
  \chapter{LA RECTA}
  \section{Ecuación de la recta}
  \subsection{Forma punto pendiente de la recta}
  \section{Distancia entre entre dos rectas}
  \subsection{Subseción de prueba}
\end{document}
```



3.10 Paquete titlesec

Referencia rápida

Herramientas

El comando `\titlelabel` permite cambiar el formato de label o etiqueta de secciones y subsecciones.

`\titlelabel`

```
\titlelabel{<label-format>}
```

Ejemplo:

Ejemplo 3.11

```
\documentclass{book}
\usepackage[paperwidth=17cm, paperheight=17cm, margin=5mm]{geometry}
\usepackage{titlesec}
\titlelabel{\thetitle -- }
\begin{document}
  \chapter{LA RECTA}
  \section{Ecuación de la recta}
  \subsection{Forma punto pendiente de la recta}
  Forma representación de contador section: \thesection
  \section{Distancia entre entre dos rectas}
  \subsection{Subseción de prueba}
\end{document}
```

Chapter 1

LA RECTA

1.1– Ecuación de la recta

1.1.1– Forma punto pendiente de la recta

Forma representación de contador section: 1.1

1.2– Distancia entre entre dos rectas

1.2.1– Subseción de prueba

\titleformat *

Para cambiar el formato de los capítulos, secciones y subsecciones se debe de utilizar el comando \titleformat*.

```
\titleformat*{<label-format>}
```

Ejemplo 3.12

```
\documentclass{book}
\usepackage[paperwidth=17cm, paperheight=17cm, margin=5mm]{geometry}
\usepackage{titlesec}
\titleformat*{\section}{\large\bfseries}
\titleformat*{\subsection}{\normalfont\bfseries}
\begin{document}
  \chapter{LA RECTA}
  \section{Ecuación de la recta}
  \subsection{Forma punto pendiente de la recta}
  El nombre del contador section es: \thesection
  \section{Distancia entre entre dos rectas}
  \subsection{Subseción de prueba}
\end{document}
```

Chapter 1

LA RECTA

1.1 Ecuación de la recta

1.1.1 Forma punto pendiente de la recta

El nombre del contador section es: 1.1

1.2 Distancia entre entre dos rectas

1.2.1 Subsección de prueba

Interfaz avanzada

Formato

Formas (shapes):

1. **hang**. Es de tipo section.
2. **block**. Se debe de utilizar este formato cuando el título tiene un forma de un bloque o párrafo; además.
3. **display**. Es de tipo chapter.
4. **runin**. Es de tipo paragraph.

Para cambiar los formatos de los capítulos, secciones y subsecciones debemos de utilizar el comando `\titleformat`. Sintaxis del comando `\titleformat`:

\titleformat

```
\titleformat{<command>}[<shape>]{<format>}{<label>}{<sep>}{<before-code>}[<after-code>]
```

1. `<command>`: puede adoptar los valores `\part`, `\chapter`, `\section`, `\subsection`, `\subsubsection`, `\paragraph` y `\subparagraph`.
2. `<shape>`: adoptar los valores expuestos anteriormente.
3. `<format>`: en esta sección se establece el formato del título.

4 <label>: se establece el formato del nombre del contador.

5 <sep>: separador del nombre del label (nombre de contador) y el título.

6 <before-code>: código que se ejecuta antes de la ejecución del comando <command> y si el uso del paquete si estás realizando con la opción explicit (\usepackage[explicit]{titlesec}), entonces el nombre del título se especifica por #1.

7 <after-code>: código que se ejecuta después de la ejecución del comando <command>.

Espaciado

\titlespacing *

Para administrar el espacio de los títulos se debe utilizar el comando \titlespacing*. Sintaxis del comando \titlespacing*:

```
\titlespacing*{<command>}{<left>}{<before-sep>}{<after-sep>}[<right-sep>]
```

La sección de comando puede ser reemplazado por \part, \chapter, \section, \subsection, \subsubsection, \paragraph y \subparagraph

Ejemplo de uso:

```
\titlespacing{\section}{0pt}{0pt}{0pt}
```

Asignar estilo de página

Supongamos que tenemos el título de un capítulo en una hoja completa, como sucede en los libros estadounidenses, entonces es necesario asignar a esta página un estilo, para ello existe el comando \assignpagestyle.

\assignpagestyle

Sintaxis del comando \assignpagestyle:

```
\assignpagestyle{<command>}{<page-style>}
```

Ejemplo de uso:

```
\assignpagestyle{\chapter}{empty}
```

Ejemplos de titlesec

Ejemplo 3.13

```
\documentclass{article}
\usepackage[utf8]{inputenc}
\usepackage[paperwidth=17cm, paperheight=21cm, margin=5mm]{geometry}
\usepackage{xcolor}
```

```
\usepackage[explicit]{titlesec}
\usepackage{soul}

\definecolor{titleblue}{HTML}{4a7aa4}

\title{Sections and Chapters}
\author{Educ}
\date{\today}

\newbox\TitleUnderlineTestBox
\newcommand*\TitleUnderline[1]
  {%
    \bgroup
    \setbox\TitleUnderlineTestBox\hbox{\colorbox{titleblue}\strut}%
    \setul{\dimexpr\dp\TitleUnderlineTestBox-.3ex\relax}{.3ex}%
    \ul{#1}%
    \egroup
  }
\newcommand*\SectionNumberBox[1]
  {%
    \colorbox{titleblue}
      {%
        \makebox[2.5em][c]
          {%
            \color{white}%
            \strut
            \csname the#1\endcsname
          }%
      }%
    \TitleUnderline{\ \ \ }%
  }
\titleformat{\section}
  {\Large\bfseries\sffamily\color{titleblue}}
  {\SectionNumberBox{section}}
  {0pt}
  {\TitleUnderline{#1}}
\titleformat{\subsection}
  {\large\bfseries\sffamily\color{titleblue}}
  {\SectionNumberBox{subsection}}
  {0pt}
  {\TitleUnderline{#1}}

\begin{document}

\maketitle
```

```
\section{Plya's Problem-Solving Cycle}
\subsection{Understand the problem}
\subsection{Devise a Plan}
\subsection{Carry Out the Plan}
\subsection{Look Back}
\section{Second Section}

\section
  {Really long section name that is really long, so long it takes two rows}
\end{document}
```

Sections and Chapters

Educ

June 4, 2023

1 Plya's Problem-Solving Cycle

1.1 Understand the problem

1.2 Devise a Plan

1.3 Carry Out the Plan

1.4 Look Back

2 Second Section

3 Really long section name that is really long, so long it takes two rows

Fuente del ejemplo: `https://tex.stackexchange.com/questions/429441/beautiful-section-styles`

3.11 Estructura de tabla de contenidos

TOC

A la tabla de contenidos se la conoce como TOC (Table of Contents) y es una lista que imprime las secciones y sus correspondientes númeraciones de páginas. Al momento de compilar un documento LaTeX se genera un archivo de extensión .toc y este archivo contiene esta lista de secciones. Al igual que para las secciones, se genera también otro archivo de extensión .lof para figuras y .lot para tablas.

`.toc`, `.lof` y `.lot`

La información que contienen los archivos generados en la compilación se puede imprimir con:

```
\tableofcontents
\listoffigures
\listoftables
```

En algunas ocasiones al compilar un documento LaTeX suele compilarse dos veces, esto se debe a que primero se recaba la información del documento LaTeX y en la segunda compilación se utiliza la información recabada. En algunas ocasiones suele compilarse hasta 3 veces debido a las referencias cruzadas.

\tableofcontents, \listoffigures y \listoftables

Añadir nuevos datos a tabla de contenidos

Para añadir nuevos datos a los archivos .toc, .lof y .lot existe el comando \addcontentsline.

```
\addcontentsline{<ext>}{<type>}{<text>}
```

El argumento <ext> puede adoptar los valores toc, lof o lot, dependiendo del archivo al cual se desea agregar. El argumento <type> se debe reemplazar por el tipo de entrada como \section, \subsection y otros. El argumento <text> se debe reemplazar por un texto, y los comandos se deben proteger con el comando \protect

El agregado de datos a tabla de contenidos se realiza de manera automática cuando se invoca algún comando como \part, \section o \subsection y también se agrega de manera automática cuando se invoca el comando \caption. Como el comando \addcontentsline solo tiene el argumento para texto, entonces es necesario añadir la representación del contador como argumento en la sección <text> y esto se logra realizando la siguiente operación:

```
\protect\numberline{<number>}<text heading>
```

Por ejemplo, el comando \caption internamente dentro del entorno \figure añade los datos a tabla de contenido de la siguiente forma:

```
\addcontentsline{lof}{figure}{%
  \protect\numberline{\thefigure}<caption text>%
}
```

Gracias al comando \protect el comando \numberline se guarda en el archivo externo y mientras tanto el comando \thefigure se expande. Posteriormente por medio del comando \numberline podemos formatearlo.

Ejemplo 3.14

```
\documentclass{article}
\usepackage{geometry}
\geometry{left=5mm,right=5mm,top=5mm,bottom=5mm,paperwidth=14cm,paperheight=10cm}
\begin{document}
  \tableofcontents
```

```
  \section*{Sección con estrella}
  Esta sección no entra en tabla de contenido, por ello, para adicionar debemos utilizar el comando
  ↪  addcontentsline
  \section{Sección sin estrella}
  \subsection{Subsección de prueba}
  \subsubsection{Subsubsección de prueba}
\end{document}
```

Contents

Sección con estrella

Esta sección no entra en tabla de contenido, por ello, para adicionar debemos utilizar el comando addcontentsline

1 Sección sin estrella

1.1 Subsección de prueba

1.1.1 Subsubsección de prueba

Ejemplo 3.15

```
\documentclass{article}
\usepackage{geometry}
\geometry{left=5mm,right=5mm,top=5mm,bottom=5mm,paperwidth=14cm,paperheight=10cm}
\begin{document}
  \tableofcontents
  \section*{Sección con estrella}
  \addcontentsline{toc}{section}{\protect\numberline{}Sección con estrella}
  \section{Sección sin estrella}
  \subsection{Subsección de prueba}
  \subsubsection{Subsubsección de prueba}
  \cite{b1}
  \begin{thebibliography}{9}
    \addcontentsline{toc}{section}{\refname}
    \bibitem{b1} Charles Lehman, \emph{Geometría analítica}, 1996
  \end{thebibliography}
\end{document}
```

Contents

Sección con estrella

1 Sección sin estrella

1.1 Subsección de prueba

1.1.1 Subsubsección de prueba

[1]

References

[1] Charles Lehman, *Geometría analítica*, 1996

Paquete titletoc

Este paquete es un complemento del paquete `titlesec` por lo que comparte la misma filosofía.

Guía rápida de paquete titletoc

El paquete `titletoc` se proporciona a los siguientes comandos:

1. `\contentslabel{<length>}`, este comando nos permite crear un espacio e imprimir el número de sección.
2. `\contentspage`, este comando crear un espacio a la derecha e imprime el número de la página.

`\contentslabel`

`\contentspage`

`\dottedcontents`, `\titlecontents` y `\contentsmargin`: son los comandos fundamentales proporcioados por el paquete `titletoc`.

Sintaxis del comando `\dottedcontents`:

`\dottedcontents`

```
\dottedcontents{<section>}[<left>]{<above code>}
                {<label width>}{<leader width>}
```

En donde,

section es el nombre de la sección, como `part`, `chapter`, `section`, `figure` y `table`. Se pueden utilizar los comandos como `\filleft`, `\filright`, `\filcenter` y `\fillast` proporcionados por el paquete `titlesec`.

left es el espacio en el margen izquierdo medido desde el margen izquierdo de la página.

label width es el ancho del espacio para el número de sección.

leader width es el ancho del espacio para el número de página.

\titlecontents

Sintaxis del comando \titlecontents:

```
\titlecontents{<section>}[<left>]{<above code>}
              {<numbered-entry-format>}{<numberless-entry-format>}
              {<filler-page-format>}[<below-code>]
```

En donde,

numbered entry format es un argumento en modo horizontal que se utilizará justo antes del título de la entrada.

numberless entry format es similar al anterior, pero si no se proporciona un valor, no habrá una etiqueta.

filler page format según la experiencia, se debe proporcionar la enumeración de la página de sección.

below code es el código que se ejecuta justo después de la etiqueta de la sección.

Los argumentos <section>, <left> y <above code> son los mismos que en el comando \dottedcontents.

\contentsmargin

Sintaxis del comando \contentsmargin:

```
\contentsmargin{<right>}
```

El comando \contentsmargin permite cambiar el margen derecho de la tabla de contenido. El valor establecido se utiliza en todas las secciones.

Ejemplo 3.16

```
\documentclass{book}
\usepackage[x11names,table,svgnames,usenames,dvipsnames]{xcolor}
\definecolor{b4}{HTML}{2196F3}
\usepackage{titlesec,titletoc,geometry,tikz}
\geometry{paperwidth=13cm,paperheight=13cm,margin=2cm}
%
\contentsmargin{0cm}
%estilo de chapter
\titlecontents{chapter}[2cm]{
  \addvspace{20pt}
  \begin{tikzpicture}[remember picture, overlay]
      \path[fill=b4] (-4.2,-.1) rectangle (-0.4,.55);
      \pgftext[left,x=-3.5cm,y=2.5mm]{\color{white}\normalsize\bfseries CAPÍTULO\; \thecontentslabel};
  \end{tikzpicture}\color{b4}\normalsize\bfseries
}{}{}{%
  \;\titlerule\;\normalsize\bfseries \thecontentspage
```

```
}
% Estilo de sección
\titlecontents{section}[3.5cm]
{\addvspace{5pt}\sffamily\bfseries}
{\contentslabel[\thecontentslabel]{1.25cm}}
{}
{\sffamily\hfill\color{black}\thecontentspage}
[]
% Estilo  Subsección
\titlecontents{subsection}[3.8cm]
{\addvspace{1pt}\sffamily\small}
{}
{}
{\sffamily\;\titlerule*[.5pc]{.}\;\thecontentspage}
[]
\begin{document}
\tableofcontents
\chapter{Capítulo 1}
\section{Sección 1}
\subsection{Subsección 1}
\section{Sección 2}
\subsection{Subsección 2}
\end{document}
```

Contents

1

2 *CONTENTS*

3

4 CHAPTER 1. CAPÍTULO 1

Chapter 1

Capítulo 1

1.1 Sección 1

1.1.1 Subsección 1

1.2 Sección 2

1.2.1 Subsección 2

Insertar una figura en contenido

Para incluir una figura en la tabla de contenido debermos recurrir al comando `\addtocontents`.

```
\necommand\figuretoc[#1]{\includegraphics{#1}}
\addtocontents{toc}{\protect\figuretoc{mifig.png}}
```

3.12 Referencias en el documento LaTeX

`\label`
`\ref`
`\pageref`

Se denomina referencia cruzada a las referencias internas entre elementos en un documento LaTeX.

Comandos para el manejo de referencias cruzadas: y

```
\label{<key>}
\ref{<key>}
\pageref{<key>}
```

El comando `\label` asigna el key al elemento actual activo. El comando `\ref` genera una referencia cruzada a un elemento con el key. El comando `\pageref` genera una referencia cruzada a la página en la que se encuentra el key.

Referencias en secciones

Ejemplo 3.17

```
\documentclass{article}
\usepackage[paperwidth=12cm,paperheight=5cm,margin=5mm]{geometry}
\begin{document}
  \section{Conjunto de números reales}\label{sec:numerosReales}
  El sección \ref{sec:numerosReales} explica acerca del conjunto de números reales.
\end{document}
```

1 Conjunto de números reales

El sección 1 explica acerca del conjunto de números reales.

HERRAMIENTAS DE FORMATO

4.1 Formato de texto

Caracteres especiales de LaTeX

En LaTeX existen ciertos caracteres especiales como las llaves, el signo dólar, el ampersand y otros caracteres que tienen una función definida; por lo tanto, es necesario estudiar a detalle estos caracteres especiales. Si se quiere imprimir un carácter especial se debe recurrir a ciertos entornos y comandos como

Tabla 4.4 Lista de caracteres especiales

Caracter	
\	Inicializa un comando.
{}	Delimita un bloque de código.
$	Inicializa y finaliza el modo matemático.
&	Este carácter es utilizado para realizar tablas y matrices.
#	Indica el parámetro de un macro. Es utilizado cuando se crea macros.
_ ^	Indica el subíndice y exponente, y es utilizado en modo matemático.
~	Este carácter añade un espacio en blanco.
%	Se utiliza para comentar código LaTeX.

`verbatim`. Supongamos que escribimos los siguientes caracteres,

Imprimir caracteres especiales

```
# \ _ ^
```

en un documento LaTeX. Al momento de compilar este documento LaTeX obtendremos una serie de errores debido a que estos caracteres tienen una función definida y no se están utilizando correctamente, por ello, para imprimir caracteres especiales en algunos casos se antepone el carácter backslash " \" y en algunos casos se recurre a otros comandos como `\verb@<code>@` o entornos proporcionados por paquetes. Para elaborar este libro se ha utilizado el paquete minted.

\verb

Para imprimir el código en línea como se observa en la pantalla debemos de utilizar el comando `\verb@<code>@`.

Ejemplo 4.1

```
\verb@$\frac{a}{b} + \sqrt{\frac{a}{b}}$@
```

$\frac{a}{b} + \sqrt{\frac{a}{b}}$

También podemos utilizar el entorno `minted` proporcionado por el paquete `minted`.

Tamaño de letras

Comando	Produce
`\tiny tiny`	tiny
`\scriptsize scriptsize`	scriptsize
`\footnotesize footnotesize`	footnotesize
`\small small`	small
`\normalsize normalsize`	normalsize
`\large large`	large
`\Large Large`	Large
`\huge huge`	huge
`\Huge Huge`	Huge

Tabla 4.6

Tamaños de letras

Ejemplo de aplicación:

Ejemplo 4.2

```
\Large\bfseries Hola mundo.
```

Hola mundo.

Ejemplo 4.3

```
\textbf{\Large Hola mundo.}
```

Hola mundo.

Mayúsculas y minúsculas

\MakeUppercase

\MakeLowercase

Los comandos:

```
\MakeUppercase{<text>}
\MakeLowercase{<text>}
```

Convierten el texto a mayúsculas y minúsculas.

Ejemplo 4.4

```
\MakeUppercase{Texto a convertir a mayúscula}\\
\MakeLowercase{TEXTO A CONVERTIR EN MINÚSCULA}
```

TEXTO A CONVERTIR A MAYÚSCULA
texto a convertir en minúscula

Expandir macro con \MakeUppercase

Estos comandos también pueden expandir macros, siempre y cuando esté definido el macro a expandir.

```
\newcommand{\mymacro}{contenido del macro}
\MakeUppercase{\mymacro texto a convertir a mayúscula}
```

No se recomienda utilizar este comando para expresiones matemáticas ni para los comandos como `\label`, `\ref` y `\cite`.

Los problemas anteriores (específicamente las referencias cruzadas y fórmulas matemáticas) se resuelven utilizando el paquete `textcase`. Este paquete proporciona los comandos `\MakeTextUppercase` y `\MakeTextLowercase`.

\MakeTextUppercase

\MakeTextLowercase

\emph

Subrayado con el paquete ulem

El paquete `ulem` redefine el comando `\emph`. Cada palabra se contiene en una caja por lo que una palabra subrayada no se puede romper automáticamente.

Ejemplo 4.5

```
\emph{Hola mundo, hola mundo, hola mundo, hola mundo...}
```

Hola mundo, hola mundo, hola mundo, hola mundo...

Otra forma:

Ejemplo 4.6

```
\uuline{Hola mundo}\\
\uwave{Hola mundo}\\
\sout{Hola mundo}\\
\xout{Hola mundo}
```

Hola mundo
Hola mundo
~~Hola mundo~~
Hola mundo

Direcciones url

\url

Una dirección URL de una página web suele contener caracteres especiales de LaTeX; por lo tanto, para imprimir una dirección URL en LaTeX se debe utilizar el comando `\url` que proporciona el paquete `url`.

Ejemplo 4.7

```
\url{http://www.google.com}
```

http://www.google.com

Para cambiar el estilo de una URL, se debe utilizar el comando `\urlstyle`.

```
\urlstyle{<style>(default tt)}
```

Los estilos predefinidos son: `rm`, `sf`, `tt` y `same`. Los tres primeros estilos hacen referencia a la familia de fuentes, correspondientemente.

Ejemplo 4.8

```
\urlstyle{sf}\url{http://www.google.com}\\
\urlstyle{same}\url{http://www.google.com}
```

http://www.google.com
http://www.google.com

Ajuste de texto en LaTeX

Los párrafos en LaTeX están formateados bajo un algoritmo que está implementado en el sistema TeX. Las líneas de un párrafo se ajustan de tal modo que tengan la misma longitud; por ello, en algunas ocasiones se estiran o se aplastan entre palabras.

flushright

flushleft

Alinear texto

En LaTeX para alinear un texto a la derecha y a la izquierda existen los entornos `flushright` y `flushleft`, respectivamente.

Ejemplo 4.9

```
\begin{flushright}
  Hola mundo
\end{flushright}
\begin{flushleft}
  Hola mundo
\end{flushleft}
\begin{center}
  Hola mundo
\end{center}
```

Hola mundo

Hola mundo

Hola mundo

\raggedleft

\raggedright

\baselineskip

\baselinestretch

También existen los comandos `\centering`, `\raggedleft` y `\raggedright` para alinear textos, pero a diferencia de los entornos flushleft, flushright y center, estos no crean espacios verticales ni inician un nuevo párrafo. Los comandos mencionados se recomiendan utilizar dentro de los entornos como `parbox`, `array`, `tabular` y `minipage`. El comando `\centering` suele utilizarse dentro del entorno `figure`.

Setspace

La longitud `\baselineskip` de TeX mide la distancia entre las líneas base de dos líneas de texto. Para cambiar la distancia entre líneas de texto, no se recomienda modificar globalmente `\baselineskip`. En LaTeX existe `\baselinestretch` para cambiar `\baselineskip` globalmente.

Ejemplo 4.10

```
\documentclass{book}
\usepackage{geometry}
\geometry{paperwidth=11cm,paperheight=16cm,margin=1cm}
\usepackage{lipsum}
\renewcommand\baselinestretch{1.5}
\begin{document}
\lipsum[1]
\end{document}
```

1

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

El paquete `setspace` proporciona entornos y comandos para modificar el espaciado entre líneas de texto. Los comandos `\singlespacing`, `\onehalfspacing` y `\doublespacing` especifican la distancia entre líneas de texto y se deben declarar en el preámbulo del documento LaTeX. Para especificar otro valor de la distancia, existe el comando `\setstretch`.

setspace

Ejemplo 4.11

```
\documentclass{book}
\usepackage{geometry}
\geometry{paperwidth=11cm,paperheight=11cm,margin=1cm}
\usepackage{lipsum}
\usepackage{setspace}
\onehalfspacing
\begin{document}
\lipsum[1]
\end{document}
```

1

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

En LaTeX, al igual que en Microsoft Word, se pueden colocar notas en pie de página. En LaTeX se complica colocar las notas en pie de página cuando se tiene un documento de 2 columnas, pero existen ciertos paquetes que facilitan la inserción de notas en pie de página. En este caso, cuando se tiene un documento de 2 columnas, se colocan en la parte inferior de la columna derecha.

Notas en pie de página por medio del comando LaTeX(estándar)

`\footnote`

LaTeX tiene la posibilidad de identificar una nota en pie de página, texto principal y nota en pie del entorno minipage. Las notas en pie de página se enumeran (utilizando el comando `\footnote` fuera del entorno minipage) y mientras tanto, en el entorno minipage (dentro del entorno minipage) se utiliza el contador `mpfootnote`.

Ejemplo 4.12

```
\documentclass{book}
\usepackage{geometry}
\geometry{paperwidth=11cm,paperheight=12cm,margin=1cm}
\usepackage{lipsum}
\begin{document}
\begin{minipage}{\textwidth}
  Contenido dentro del entorno minipage. \footnote{nota en pie de página dentro del entorno minipage}
\end{minipage}

Prueba de nota en pie de página \footnote{Nota en pie de página fuera del entorno minipage}
```

```
Contador footnote: \thefootnote\\
Contador mpfootnote: \thempfootnote
\end{document}
```

1

Contenido dentro del entorno minipage. [a]

[a]nota en pie de página dentro del entorno minipage

Prueba de nota en pie de página [1]

Contador footnote: 1

Contador mpfootnote: *a*

[1]Nota en pie de página fuera del entorno minipage

Hay la posibilidad de cambiar la representación de los contadores `footnote` y `mpfootnote`.

Ejemplo 4.13

```
\documentclass{book}
\usepackage{geometry}
\geometry{paperwidth=11cm,paperheight=12cm,margin=1cm}
\usepackage{lipsum}
\renewcommand\thefootnote{\fnsymbol{footnote}}
\renewcommand\thempfootnote{\Alph{mpfootnote}}
\begin{document}
\begin{minipage}{\textwidth}
  Contenido dentro del entorno minipage. \footnote{Nota en pie de página dentro del entorno minipage}
\end{minipage}
Primera prueba de nota en pie de página \footnote{Nota en pie de página fuera del entorno minipage}.

Segunda prueba de nota en pie de página \footnote{Nota en pie de página fuera del entorno minipage}.

Contador footnote: \thefootnote\\
Contador mpfootnote: \thempfootnote
\end{document}
```

1

Contenido dentro del entorno minipage. [A]

[A]Nota en pie de página dentro del entorno minipage

Primera prueba de nota en pie de página *.

Segunda prueba de nota en pie de página †.

Contador footnote: †

Contador mpfootnote: A

*Nota en pie de página fuera del entorno minipage

†Nota en pie de página fuera del entorno minipage

Estilos de numeración de notas en pie de página:

1. alph
2. Alph
3. arabic
4. fnsymbol
5. Roman

`\footnotemark`

`\footnotetext`

En el ejemplo podemos apreciar que al invocar el comando `\footnote` dentro del entorno minipage, la nota se imprime en el pie del cuadro. Si se desea imprimir la nota en el pie de página principal, se debe recurrir a los comandos `\footnotemark` y `\footnotetext`. El comando `\footnotemark` debe ir dentro del entorno minipage y el comando `\footnotetext` después del entorno minipage.

Ejemplo 4.14

```
\documentclass{book}
\usepackage{geometry}
\geometry{paperwidth=11cm,paperheight=12cm,margin=1cm}
\begin{document}
\begin{minipage}{\textwidth}
  Contenido dentro del entorno minipage. \footnotemark{}
\end{minipage}\footnotetext{Nota en pie de página}
\end{document}
```

1

Contenido dentro del entorno minipage. [1]

[1] Nota en pie de página

4.2 Notas en el margen

Internamente, LaTeX procesa las notas en el margen como objetos flotantes.

\marginpar

El comando \marginpar es utilizado para colocar cualquier texto o imagen en el margen y es un comando de LaTeX. El texto aparece justo a la misma altura donde se invoca el comando. Se debe tener en cuenta que el comando \marginpar no inicia un nuevo párrafo.

Ejemplo 4.15

```
\documentclass{book}
\usepackage{geometry}
\geometry{%
  includemp,
  paperwidth=14cm,paperheight=12cm,margin=1cm,
  marginparsep=2mm, marginparwidth=3cm,
  showframe
}
\usepackage{lipsum}
\begin{document}
\lipsum[1]
\marginpar{Texto en el margen}
\end{document}
```

1

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

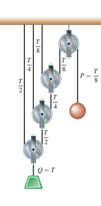
Texto en el margen

Ejemplo 4.16

```
\documentclass{book}
\usepackage{geometry}
\usepackage{graphicx}
\geometry{%
  includemp,
  paperwidth=14cm,paperheight=12cm,margin=1cm,
  marginparsep=2mm, marginparwidth=3cm,
  showframe
}
\usepackage{lipsum}
\begin{document}
\marginpar{\includegraphics[width = \marginparwidth]{images/grafica1}}
\lipsum[1]
\end{document}
```

1

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.



Cuando se utiliza varias veces el comando `\marginpar`, suele desordenarse la disposición de las notas en el margen, es decir, el texto en el margen no aparece en la misma altura. Para solucionar este problema, se puede utilizar el paquete `mparhack`.

mparhack

marginfix

Cuando se tienen márgenes largos, estos suelen sobresalir de la página. Para solucionar este problema, Stephen Hicks ha creado el paquete `marginfix`. Cuando se utiliza `marginfix`, no es necesario utilizar el paquete `mparhack`.

`\blockmargin` y `\unblockmargin`

El comando `\blockmargin` y `tex` permite bloquear el margen izquierdo y derecho respectivamente. Estos comandos no se pueden invocar dentro de un entorno flotante, en vez de ello podemos invocar el comando blockmargin y unblockmargin antes y después del entorno flotante siempre y cuando el entorno flotante tenga la opción `[H]` como argumento opcional. Esta técnica es utilizado cuando las figuras ocupan todo el ancho del papel (`\paperwidth`).

Ejemplo 4.17

```
\documentclass[twoside]{book}
\usepackage{lipsum,marginfix,float}
\usepackage{geometry}
\geometry{%
  includehead, includefoot,includemp,heightrounded,
  paperwidth=20cm,left=1.5cm,right=1cm, marginparsep=5mm, marginparwidth=5cm,
  paperheight=20cm,top=8mm,bottom=8mm,
}
\makeatletter
\begingroup
\catcode`\Q=3
  \long\gdef\@ifmtarg#1{\@xifmtarg#1QQ\@secondoftwo\@firstoftwo\@nil}
  \long\gdef\@xifmtarg#1#2Q#3#4#5\@nil{#4}
```

```
\endgroup
\newenvironment{adjustwidth*}[2]{%
\begin{list}{}{%
  \topsep\z@%
  \listparindent\parindent%
  \parsep\parskip%
  \ifodd\c@page%     odd numbered page
    \@ifmtarg{#1}{\setlength{\leftmargin}{\z@}}%
                 {\setlength{\leftmargin}{#1}}%
    \@ifmtarg{#2}{\setlength{\rightmargin}{\z@}}%
                 {\setlength{\rightmargin}{#2}}%
  \else%           even numbered page
    \@ifmtarg{#2}{\setlength{\leftmargin}{\z@}}%
                 {\setlength{\leftmargin}{#2}}%
    \@ifmtarg{#1}{\setlength{\rightmargin}{\z@}}%
                 {\setlength{\rightmargin}{#1}}%
  \fi}
  \item[]}{\end{list}}
\makeatother
\begin{document}
\lipsum[1]
\marginpar{Texto en el margen antes de invocar entorno flotante}
\blockmargin
\begin{figure}[H]
  \begin{adjustwidth*}{}{\dimexpr-\marginparsep-\marginparwidth}
    \lipsum[1]%reemplazar con algún imagen
  \end{adjustwidth*}
\end{figure}
\unblockmargin
\marginpar{Texto en el margen después de invocar entorno flotante}
\lipsum[1]
\end{document}
```

1

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Texto en el margen antes de invocar entorno flotante

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Texto en el margen después de invocar entorno flotante

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Si el argumento opcional de los entornos flotantes está establecido a `tb`, entonces hace un bloqueo automático de las notas en margen en el lugar en donde aparece (en pdf) el flotante. Esta operación generalmente suele utilizarse cuando el flotante se extiende hasta la parte de notas en el margen.

Ejemplo 4.18

```
\documentclass[twoside]{book}
\usepackage{lipsum,marginfix,float}
\usepackage{geometry}
\geometry{%
  includehead, includefoot,includemp,heightrounded,
  paperwidth=20cm,left=1.5cm,right=1cm, marginparsep=5mm, marginparwidth=5cm,
  paperheight=20cm,top=8mm,bottom=8mm,
}
\makeatletter
\begingroup
\catcode`\Q=3
  \long\gdef\@ifmtarg#1{\@xifmtarg#1QQ\@secondoftwo\@firstoftwo\@nil}
  \long\gdef\@xifmtarg#1#2Q#3#4#5\@nil{#4}
\endgroup
\newenvironment{adjustwidth*}[2]{%
\begin{list}{}{%
  \topsep\z@%
  \listparindent\parindent%
  \parsep\parskip%
  \ifodd\c@page%     odd numbered page
```

```
    \@ifmtarg{#1}{\setlength{\leftmargin}{\z@}}%
                 {\setlength{\leftmargin}{#1}}%
    \@ifmtarg{#2}{\setlength{\rightmargin}{\z@}}%
                 {\setlength{\rightmargin}{#2}}%
  \else%           even numbered page
    \@ifmtarg{#2}{\setlength{\leftmargin}{\z@}}%
                 {\setlength{\leftmargin}{#2}}%
    \@ifmtarg{#1}{\setlength{\rightmargin}{\z@}}%
                 {\setlength{\rightmargin}{#1}}%
  \fi}
  \item[]}{\end{list}}
\makeatother
\begin{document}
\lipsum[1]
\marginpar{Texto en el margen antes de invocar entorno flotante}
\begin{figure}[tb]
  \begin{adjustwidth*}{}{\dimexpr-\marginparsep-\marginparwidth}
    \lipsum[1]%reemplazar con algún imagen
  \end{adjustwidth*}
\end{figure}%
\marginpar{Texto en el margen después de invocar entorno flotante}%
\lipsum[1]
\end{document}
```

1

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Texto en el margen antes de invocar entorno flotante
Texto en el margen después de invocar entorno flotante

Personalizar notas en pie de página

LaTeX ofrece varios parámetros para personalizar el aspecto de las notas al pie. Algunos de ellos son:

1. `\footnotesize`: tamaño de fuente utilizado dentro de las notas al pie.
2. `\footnotesep`: altura de un salto (strut) colocado al comienzo de cada nota al pie. Si es mayor que el `\baselineskip` usado para `\footnotesize`, se insertará espacio vertical adicional sobre cada nota al pie. `\skip\footins`: parámetro que define el espacio entre el texto principal y el inicio de las notas al pie. Puedes cambiar su valor con el comando `\setlength` o `\addtolength`. Ejemplo:

```
\addtolength{\skip\footins}{10mm plus 2mm}
```

3. `\footnoterule`: comando que define la línea horizontal que separa el texto principal de las notas al pie. Por defecto es una línea horizontal de 0.4pt de grosor. Puedes cambiar su valor con el comando `\renewcommand`. Ejemplo:

```
\renewcommand\footnoterule{\vspace*{-3pt}\hrule width 2in height 0.4pt
↪  \vspace*{2.6pt}}
```

\footnoterule \skip\footins
\footnotesep
1 \@makefntext
producido por \@makefnmark
\footnotesep
2 \@makefntext
producido por \@makefnmark

Figura 4.1 Esquema de la nota de pie de página

En la clase article, las numeraciones de notas en pie de página se pueden resetear con el comando `\@addtoreset` y mientras en las clases report y book las numeraciones notas en pie se resetean cada vez que se invoca el comando `\chapter`.

Comando `\@makefnmark`: Este comando genera el marcador de la nota al pie. Por defecto, crea un marcador en superíndice. Su definición es la siguiente:

```
\renewcommand\@makefnmark{\mbox{\textsuperscript{\normalfont\@thefnmark}}}
```

Comando `\@makefntext`: Este comando se ejecuta dentro de un entorno `\parbox` con un ancho de `\columnwidth` y define la apariencia de la nota al pie. La versión predeterminada se muestra a continuación:

```
\newcommand\@makefntext[1]{\noindent\makebox[1.8em][r]{\@makefnmark}#1}
```

Paquete sidenote

El paquete `sidenotes` también nos permite poner notas en el margen, como texto, figuras y tablas.

description

4.3 Cajas

mbox

\mbox es un comando que nos permite crear una caja de texto y con bordes invisibles.

Ejemplo 4.19

```
\mbox{Texto en una caja}
```

Texto en una caja

fbox

\fbox es un comando que nos permite crear una caja de texto con bordes visibles.

Ejemplo 4.20

```
\fbox{Texto en una caja}
```

Texto en una caja

makebox

Este comando crea caja con bordes invesible y su sintaxis es: \makebox[ancho][justificacion␣]{contenido}. Las justificaciones pueden ser: l (izquierda), r (derecha), c (centro) y s (estirado).

Ejemplo 4.21

```
\makebox[5cm][l]{Texto en una caja}\\
\makebox[5cm][r]{Texto en una caja}\\
\makebox[5cm][c]{Texto en una caja}\\
\makebox[5cm][s]{Texto en una caja}
```

Texto en una caja
Texto en una caja
Texto en una caja
Texto en una caja

framebox

Este comando crea caja con bordes visibles y su sintaxis es: \framebox[ancho][justificacion]␣{contenido}. Las justificaciones son los mismos que de \makebox.

Ejemplo 4.22

```
\framebox[5cm][l]{Texto en una caja}\\
\framebox[5cm][r]{Texto en una caja}\\
\framebox[5cm][c]{Texto en una caja}\\
\framebox[5cm][s]{Texto en una caja}
```

Texto en una caja
Texto en una caja
Texto en una caja
Texto en una caja

Parámetros de cajas con líneas

Los parámetros de cajas con líneas como framebox y fbox son: \fboxrule, define grosor de la línea; \fboxsep, define la distancia entre línea y el texto de la caja.

Ejemplo 4.23

```
{\setlength\fboxrule{.5pt}\setlength\fboxsep{1pt}\fbox{Texto en una caja}}
```

Texto en una caja

parbox

Este comando generalmente es utilizado cuando se contiene párrafos en cajas (los párrafos no tienen sangría). Sintaxis:

```
\parbox[posición][altura][posición interior]{ancho}{contenido o texto}
```

En donde:

1. texto: es el texto que se va a colocar en la caja.
2. ancho: es el ancho horizontal de la caja.
3. posición: es la posición de la caja con respecto al texto que rodea a la caja. Puede ser t (arriba), c (centro) y b (abajo).
4. altura: es la altura de la caja.
5. posición interior: es la posición del texto dentro de la caja. Puede ser t (arriba), c (centro) y b (abajo).

Ejemplo 4.24

```
\begin{center}
  Texto exterior 1\\
  Texto exterior 2
  \parbox{4cm}{Praesent a rhoncus dui, id venenatis ex. Morbi dictum viverra lectus sed efficitur.
   ↪  Nullam vulputate euismod purus, eu lacinia tortor bibendum nec.}
  \hspace{5mm}
  \parbox{4cm}{In sollicitudin justo nunc, eget condimentum massa ultricies ut. Vestibulum id justo sed
   ↪  sem hendrerit volutpat at et sem. Nulla sagittis arcu libero, ut hendrerit nulla consequat id.
   ↪  Nunc ac leo ligula. Proin tempor rutrum lacus.}
  Texto exterior 3
\end{center}
```

Texto exterior 1

Texto exterior 2 Praesent a rhoncus dui, id venenatis ex. Morbi dictum viverra lectus sed efficitur. Nullam vulputate euismod purus, eu lacinia tortor bibendum nec. In sollicitudin justo nunc, eget condimentum massa ultricies ut. Vestibulum id justo sed sem hendrerit volutpat at et sem. Nulla sagittis arcu libero, ut hendrerit nulla consequat id. Nunc ac leo ligula. Proin tempor rutrum lacus. Texto exterior 3

Ejemplo 4.25

```
\begin{center}
  Texto exterior 1\\
  Texto exterior 2
  \parbox[t]{4cm}{Praesent a rhoncus dui, id venenatis ex. Morbi dictum viverra lectus sed efficitur.
  ↪  Nullam vulputate euismod purus, eu lacinia tortor bibendum nec.}
  \hspace{5mm}
  \parbox[t]{4cm}{In sollicitudin justo nunc, eget condimentum massa ultricies ut. Vestibulum id justo
  ↪  sed sem hendrerit volutpat at et sem. Nulla sagittis arcu libero, ut hendrerit nulla consequat
  ↪  id. Nunc ac leo ligula. Proin tempor rutrum lacus.}
  Texto exterior 3
\end{center}
```

Texto exterior 1

Texto exterior 2 Praesent a rhoncus dui, id venenatis ex. Morbi dictum viverra lectus sed efficitur. Nullam vulputate euismod purus, eu lacinia tortor bibendum nec. In sollicitudin justo nunc, eget condimentum massa ultricies ut. Vestibulum id justo sed sem hendrerit volutpat at et sem. Nulla sagittis arcu libero, ut hendrerit nulla consequat id. Nunc ac leo ligula. Proin tempor rutrum lacus. Texto exterior 3

Ejemplo 4.26

```
\begin{center}
  \fbox{\parbox[c][4cm][t]{3cm}{Praesent a rhoncus dui, id venenatis ex. Morbi dictum viverra lectus
  ↪  sed efficitur}}\;
  \fbox{\parbox[c][4cm][c]{3cm}{Praesent a rhoncus dui, id venenatis ex. Morbi dictum viverra lectus
  ↪  sed efficitur}}\;
  \fbox{\parbox[c][4cm][b]{3cm}{Praesent a rhoncus dui, id venenatis ex. Morbi dictum viverra lectus
  ↪  sed efficitur}}
\end{center}
```

Praesent a rhoncus dui, id venenatis ex. Morbi dictum viverra lectus sed efficitur	Praesent a rhoncus dui, id venenatis ex. Morbi dictum viverra lectus sed efficitur	Praesent a rhoncus dui, id venenatis ex. Morbi dictum viverra lectus sed efficitur

Moviendo cajas

Para mover una caja se utiliza el comando `raisebox` y su sintaxis es: `\raisebox{desplazamiento vertical}{texto}`. El desplazamiento vertical puede ser positivo o negativo.

Ejemplo 4.27

```
Hola 1 \raisebox{2em}{Hola 2} Hola 3 \raisebox{-2em}{Hola 4}
```

Hola 2
Hola 1 Hola 3
Hola 4

minipage

El entorno `minipage` se utiliza para textos más largos con respecto a `parbox`.

Sintaxis del entorno minipage:

```
\begin{minipage}[posición][altura][posición interior]{ancho}
  texto
\end{minipage}
```

Se debe tomar en cuenta que en el entorno minipage se puede edicón de página normal como incluyendo notas al pie de página. Dentro de este entorno se puede utilzar crear gráfica y tablas; por ejemplo, en las notas en el margen se puede incluir las gráficas por medio de este entorno.

Ejemplo 4.28

```
\begin{minipage}[t]{3cm}
  Praesent a rhoncus dui, id venenatis ex. Morbi dictum
   ↪ viverra lectus sed efficitur
\end{minipage}\;
\begin{minipage}[t]{3cm}
  Praesent a rhoncus dui, id venenatis ex. Morbi dictum
   ↪ viverra lectus sed efficitur
\end{minipage}
```

Praesent a rhoncus dui, id venenatis ex. Morbi dictum viverra lectus sed efficitur

Praesent a rhoncus dui, id venenatis ex. Morbi dictum viverra lectus sed efficitur

4.4 Listas

En LaTeX existen tres entornos para realizar listas, que son: `itemize`, `enumerate` y `description`.

Personalizar entorno itemize

Entorno itemize por defecto:

Ejemplo 4.29

```
\documentclass{book}
\usepackage{geometry}
\geometry{paperwidth=11cm,paperheight=7cm,margin=1cm}
\usepackage{lipsum}
\begin{document}
\begin{itemize}
  \item Item 1
  \item Item 2
  \item Item 3
    \begin{itemize}
      \item Subitem 1
      \item Subitem 2
        \begin{itemize}
          \item Subsubitem 1
          \item Subsubitem 2
        \end{itemize}
      \item Subitem 3
    \end{itemize}
\end{itemize}
\end{document}
```

1

- Item 1
- Item 2
- Item 3
  – Subitem 1
  – Subitem 2
    ∗ Subsubitem 1
    ∗ Subsubitem 2
  – Subitem 3

Para cambiar la enumeración debemos redefinir el comando `\labelitemi`.

Ejemplo 4.30

```
\renewcommand\labelitemi{\textbullet}
\begin{itemize}
  \item Item 1
  \item Item 2
  \item Item 3
\end{itemize}
```

- Item 1
- Item 2
- Item 3

También podemos cambiar las numeraciones de los subitems y subsubitems.

Personalizar entorno enumerate

enumerate

Entorno enumerate por defecto:

Ejemplo 4.31

```
\documentclass{book}
\usepackage{geometry}
\geometry{paperwidth=14cm,paperheight=8cm,margin=1cm}
\begin{document}
\begin{enumerate}
  \item Item 1
  \item Item 2
  \item Item 3
    \begin{enumerate}
      \item Subitem 1
      \item Subitem 2
        \begin{enumerate}
          \item Subsubitem 1
```

```
        \item Subsubitem 2
          \begin{enumerate}
            \item Subsubsubitem 1
          \end{enumerate}
      \end{enumerate}
    \item Subitem 3
  \end{enumerate}
\end{enumerate}
\end{document}
```

1

1. Item 1
2. Item 2
3. Item 3
   (a) Subitem 1
   (b) Subitem 2
      i. Subsubitem 1
      ii. Subsubitem 2
         A. Subsubsubitem 1
   (c) Subitem 3

itemize

Al igual que al entorno itemize, el entorno enumerate está por niveles y se resume en la tabla 4.8.

	Primer nivel	Segundo nivel	Tercer nivel	Cuarto nivel
Contador	enumi	enumii	enumiii	enumiv
Representación	\theenumi	\theenumii	\theenumiii	\theenumiv
Definición por defecto	\arabic{enumi}	\alph{enumii}	\roman{enumiii}	\Alph{enumiv}
Etiqueta	\labelenumi	\labelenumii	\labelenumiii	\labelenumiv
Forma de la etiqueta	\theenumi.	(\theenumii)	\theenumiii.	\theenumiv.

Tabla 4.8

De la tabla mostrada, podemos cambiar la forma de la etiqueta mediante la redefinición de uno de los comandos de forma de etiqueta (en este caso `\labelenumi`, `\labelenumii`, `\labelenumiii` y `\labelenumiv`). También se puede cambiar la forma de representación mediante la redefinición de uno de los comandos de representación, como `\theenumi`.

Ejemplo 4.32

```
\renewcommand\theenumi{\textbf{\arabic{enumi}}}
\renewcommand\labelenumi{%
  \tikz[baseline=-3pt]\node[shade, circle, ball
  ↪  color=Salmon!50!white]{\theenumi};
}
\begin{enumerate}
  \item Item 1
  \item Item 2
\end{enumerate}
```

1 Item 1

2 Item 2

Personalizar entorno description

Entorno description por defecto:

Ejemplo 4.33

```
\documentclass{book}
\usepackage{geometry}
\geometry{paperwidth=14cm,paperheight=8cm,margin=1cm}
\begin{document}
\begin{description}
  \item[Item 1] Item 1
  \item[Item 2] Item 2
  \item[Item 3] Item 3
    \begin{description}
      \item[Subitem 1] Subitem 1
      \item[Subitem 2] Subitem 2
        \begin{description}
          \item[Subsubitem 1] Subsubitem 1
          \item[Subsubitem 2] Subsubitem 2
        \end{description}
      \item[Subitem 3] Subitem 3
    \end{description}
\end{description}
\end{document}
```

1

Item 1 Item 1

Item 2 Item 2

Item 3 Item 3

Subitem 1 Subitem 1
Subitem 2 Subitem 2
Subsubitem 1 Subsubitem 1
Subsubitem 2 Subsubitem 2
Subitem 3 Subitem 3

En este entorno el único parámetro, relevante, a cambiar es `\descriptionlabel`.

Ejemplo 4.34

```
\renewcommand\descriptionlabel[1]{%
  \hspace{\labelsep}
  \tikz[baseline=-3pt]\node[fill=Salmon!50]{\textbf{#1}};
}
\begin{description}
  \item[Item 1] Item 1
  \item[Item 2] Item 2
  \item[Item 3] Item 3
\end{description}
```

Item 1 Item 1

Item 2 Item 2

Item 3 Item 3

¿Cómo crear listas?

list

Las listas en LaTeX son creadas con el entorno `list` y este entorno tiene la siguiente sintaxis:

```
\begin{list}{<default label>}{<decls>}
  <list elements>
\end{list}
```

En donde `<default label>` establece la etiqueta de la lista y `<decls>` establece los parámetros geométricos de la lista.

Ejemplo 4.35

```
\begin{list}{\color{red}$\heartsuit$}{}
  \item Item 1
  \item Item 2
  \item Item 3
\end{list}
```

♡ Item 1

♡ Item 2

♡ Item 3

Niveles del entorno list se muestran en la tabla 4.9.

Parámetros de una lista se puede ver en la figura 4.2.

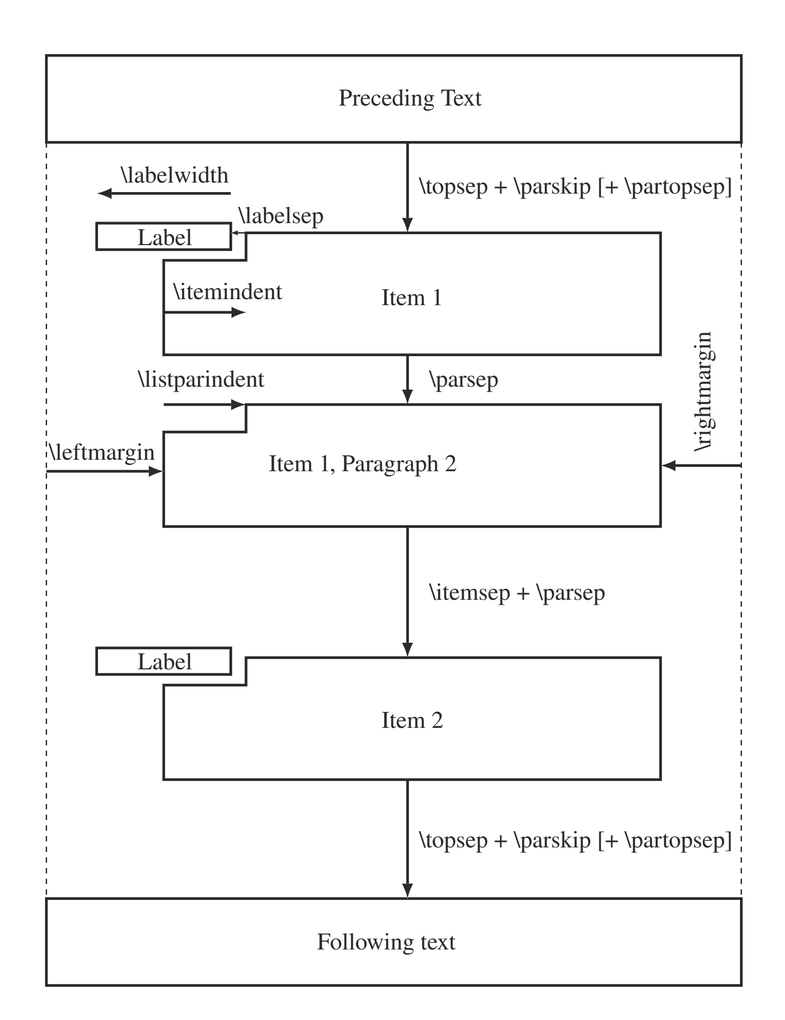


Figura 4.2 Parámetros de una lista

4.5 Paquete amsthm

Este paquete es utilizado para crear entornos de tipo teorema, pero no siempre puede ser un teorema; también se puede adaptar para un ejemplo, una definición y otros.

amsthm

El paquete `amsthm` proporciona dos comandos: `\newtheorem` y `\newtheorem*`. El comando `\newtheorem` genera una numeración y el comando `\newtheorem*` no genera enumeración.

`\newtheorem*`

Teorema sin enumeración:

```
\newtheorem*{<name>}{<heading>}
```

Ejemplo 4.36

```
\newtheorem*{lema}{Lema}
\begin{lema}
  Test.
\end{lema}
\begin{lema}[abc]
  Test.
\end{lema}
```

Lema. *Test.*

Lema (abc)**.** *Test.*

`\newtheorem` Teorema con enumeración:

```
\newtheorem{<env name>}{<text>}[<parent counter>]
\newtheorem{<env name>}[<shared counter>]{<text>}
```

El comando `\newtheorem{<name>}{<heading>}[<counter>]` crea un nuevo entorno de nombre `<name>` y su contador está subordinado al contador `<parent counter>`

Ejemplo 4.37

```
\documentclass{article}
\usepackage{amsthm,amsmath}
\usepackage{geometry}
\geometry{paperwidth=14cm,paperheight=8cm,margin=1cm}
\newtheorem{definition}{Definición}[section]
\begin{document}
\section{Sección 1}
\begin{definition}
  El desplazamiento $\Delta \vec{x}$ de una partícula se define como cambio de posición.
  \begin{align*}
    \Delta \vec{s} &= \vec{s} - \vec{s}_0
  \end{align*}
\end{definition}
\section{Sección 2}
\begin{definition}
  El desplazamiento $\Delta \vec{x}$ de una partícula se define como cambio de posición.
  \begin{align*}
    \Delta \vec{s} &= \vec{s} - \vec{s}_0
  \end{align*}
\end{definition}
\end{document}
```

1 Sección 1

Definición 1.1. *El desplazamiento $\Delta\vec{x}$ de una partícula se define como cambio de posición.*

$$\Delta\vec{s} = \vec{s} - \vec{s}_0$$

2 Sección 2

Definición 2.1. *El desplazamiento $\Delta\vec{x}$ de una partícula se define como cambio de posición.*

$$\Delta\vec{s} = \vec{s} - \vec{s}_0$$

1

La enumeración de los teoremas se resetea cada vez que el contador `section` incrementa su valor.

El comando `\newtheorem{<name>}[<counter>]{<heading>}` crea un nuevo entorno de nombre `<name>` y su contador está definido por otro teorema, definido por `\newtheorem{<name>}{<heading>}[<counter>]`.

`\newtheorem`

Ejemplo 4.38

```
\documentclass{article}
\usepackage{amsthm,amsmath}
\usepackage{geometry}
\geometry{paperwidth=14cm,paperheight=8cm,margin=1cm}
\newtheorem{thm}{Theorem}[section]
\newtheorem{defn}[thm]{Definition}
\begin{document}
\section{Sección 1}
\begin{thm}
  Test
\end{thm}
\begin{defn}
  Test
\end{defn}
\section{Sección 2}
\begin{thm}
  Test
\end{thm}
\begin{defn}
  Test
\end{defn}
\end{document}
```

1 Sección 1

Theorem 1.1. *Test*

Definition 1.2. *Test*

2 Sección 2

Theorem 2.1. *Test*

Definition 2.2. *Test*

1

Estilos de teoremas

\newtheoremstyle

Sintaxis para definir nuevo estilo de teorema:

```
\newtheoremstyle{<name style>}%
  {}% <Space above>
  {}% <Space below>
  {}% <Body font>
  {}% <Indent amount>
  {}% <Theorem head font>
  {}% <Punctuation after theorem head>
  {}% <Space after theorem head>
  {}% <Theorem head spec (can be left empty, meaning ‘normal’)>
```

Ejemplo 4.39

```
\documentclass{article}
\usepackage{amsthm,amsmath}
\usepackage{geometry}
\geometry{paperwidth=14cm,paperheight=8cm,margin=1cm}
%Definition
\newtheoremstyle{style_definition} %% Name
{}
{}
{\normalfont}
{}
{\bfseries}
{}
{4pt}
{}
\theoremstyle{style_definition}
\newtheorem{defn}{Definición}[section]
```

```
\begin{document}
\section{Sección 1}
\begin{defn}
  Test
\end{defn}
\end{document}
```

1 Sección 1

Definición 1.1 Test

1

Ejemplo 4.40

```
\documentclass{article}
\usepackage[x11names,table,svgnames,usenames,dvipsnames]{xcolor}
\usepackage{amsthm,amsmath,tcolorbox}
\tcbuselibrary{skins}
\tcbuselibrary{breakable}
\usepackage{geometry}
\geometry{paperwidth=14cm,paperheight=8cm,margin=1cm}
%Theorem
\definecolor{colbackdefinition}{HTML}{C6E8ED}%
\definecolor{ruledefinition}{HTML}{009ADA}%
\newtheoremstyle{style_theorem}{}{}{\normalfont}{}{\bfseries}{}{4pt}{}
\theoremstyle{style_theorem}
\newtheorem{defn}{Definición}[section]
\tcolorboxenvironment{defn}{%
  boxrule=0pt,
  boxsep=0pt,
  arc=3mm,
  colback={colbackdefinition},
  rounded corners=all,
  enhanced,
  before skip=10pt,
  after skip=10pt,
  left=5pt,
  right=5pt,
  fuzzy shadow={1.5pt}{-1.5pt}{0pt}{0.01pt}{black!50},
```

```
  breakable,
  if odd page*={
      overlay unbroken and last={
          \draw[line width=1pt, ruledefinition] ([yshift=-3mm,xshift=-0.5pt]frame.north east) --
          ↪ ([yshift=3mm,xshift=-0.5pt]frame.south east);
        },
    }{
      overlay unbroken and last={
          \draw[line width=1pt, ruledefinition] ([yshift=-3mm,xshift=0.5pt]frame.north west) --
          ↪ ([yshift=3mm,xshift=0.5pt]frame.south west);
        },
    },
  check odd page=true,
  width=\textwidth,
}
\begin{document}
\section{Sección 1}
\begin{defn}[Desplazamiento]
  El desplazamiento $\Delta \vec{x}$ de una partícula se define como cambio de posición.
  \begin{align*}
    \Delta \vec{s} &= \vec{s} - \vec{s}_0
  \end{align*}
\end{defn}
\end{document}
```

1 **Sección 1**

Definición 1.1 (Desplazamiento) El desplazamiento $\Delta\vec{x}$ de una partícula se define como cambio de posición.

$$\Delta\vec{s} = \vec{s} - \vec{s}_0$$

1

4.6 Texto como en la pantalla

verbatim

Para obtener el texto como se ve en la pantalla, se puede realizar de diferentes formas. LaTeX proporciona el entorno `verbatim` para este objetivo.

```
\begin{verbatim}
  Hola %^&_$$ \frac{a}{b}
\end{verbatim}
```

```
Hola %^&_$$ \frac{a}{b}
```

También existe el comando `\verb` para obtener el texto como se ve en la pantalla.

\verb

Fancyvrb

El paquete `fancyvrb` proporciona una serie de opciones para dar formato al texto como se ve en la pantalla. El entorno proporcionado por este paquete es `Verbatim`.

fancyvrb

```
\begin{Verbatim}
  Hola %^&_$$ \frac{a}{b}
\end{Verbatim}
```

```
Hola %^&_$$ \frac{a}{b}
```

Personalizar fuentes

fontfamily

Cambiando tipo de fuente:

```
\begin{Verbatim}[%
  fontfamily=helvetica
]
  Verbatim line
\end{Verbatim}
```

Verbatim line

fontsize

Cambiando tamaño de la fuente:

```
\begin{Verbatim}[fontfamily=helvetica,
                fontsize=\small]
  Verbatim line
\end{Verbatim}
```

Verbatim line

Marcos de la caja

frame

El key,

```
frame(none|leftline|topline|bottomline|lines|single)
```

crea marcos y su valor por defecto es `frame=none`.

```
\begin{Verbatim}[frame=single]
  Verbatim line
\end{Verbatim}
```

```
Verbatim line
```

Para establecer el grosor de la línea de marco se puede utilizar el key

```
framerule(dimension)
```

framerule

y el valor por defecto es `framerule=0.4pt`.

```
\begin{Verbatim}[frame=single,
                 framerule=2pt]
  Verbatim line
\end{Verbatim}
```

```
Verbatim line
```

Otras opciones:

framesep
rulecolor
fillcolor

1. framesep.
2. rulecolor.
3. fillcolor.

Etiqueta del entorno Verbatim

label

Para este objetivo se recurre al key,

```
label({[strign]string})
```

```
\begin{Verbatim}[label=Text]
  Verbatim line
\end{Verbatim}
```

```
Verbatim line
```

labelposition

Las posiciones de las etiquetas podemos establecer con el key

```
labelposition (none|topline|bottomline|all)
```

numbers

Numeración de líneas

```
numbers(none|left|right)
```

```
\begin{Verbatim}[numbers=left]
  First line
  Second line
\end{Verbatim}
```

```
1    First line
2    Second line
```

firstnumber
stepnumber
numbersep
num-berblanklines
baselinestretch

Otras opciones:

1. firstnumber (auto|last|integer).
2. stepnumber (integer).
3. numbersep (dimension).
4. numberblanklines (boolean).

Espacio entre líneas

baselinestretch (auto|dimension)

```
\begin{Verbatim}[baselinestret
↪ ch=2]
  First line
  Second line
\end{Verbatim}
```

```
First line

Second line
```

Para tener más detalle acerca de este paquete se recomienda al lector revisar su documentación oficial en http://ctan.org/pkg/verbatim.

Código de una programa

Para incluir códigos de programas en nuestro documento LaTeX, se puede realizar con el entorno verbatim, pero para leer de una manera más sencilla el código, es necesario formatearlo; por lo tanto, existen ciertos paquetes para escribir el código de un programa, como algorithmic y minted. algorithmic

```
\begin{algorithmic}
  \IF{$n$ is odd}
    \RETURN \TRUE
  \ELSE
    \RETURN \FALSE
  \ENDIF
\end{algorithmic}
```

if n is odd **then**
  return true
else
  return false
end if

En este libro se ha utilizado el paquete minted para obtener el código LaTeX.

4.7 Columnas

\twocolumn

Podemos pasar la opción de twocolumn como un argumento opcional de una clase para obtener un documento LaTeX de dos columnas.

Ejemplo 4.41

```
\documentclass[twocolumn]{article}
\usepackage{geometry}
\geometry{paperwidth=14cm,paperheight=18cm,margin=5mm}
\usepackage{lipsum}
\begin{document}
\lipsum[1-4]
\end{document}
```

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula.

También podemos utilizar el comando `\twocolumn[<prelim one column text>]` para obtener una salida de dos columnas. Para obtener un documento de una sola columna, podemos utilizar el comando `\onecolumn`.

`\onecolumn`

Parámetros de la compisicón tipográfica de dos columnas:

`\columnsep`

`\columnseprule`

1. `\columnsep`. Establece el espacio entre las columnas.
2. `\columnseprule`. Estable el grosor de la línea entre las columnas.

Ejemplo 4.42

```
\documentclass{article}
\usepackage{geometry}
\geometry{paperwidth=14cm,paperheight=18cm,margin=5mm}
\usepackage{lipsum}
\setlength{\columnsep}{1cm}
\setlength{\columnseprule}{0.5pt}
\begin{document}
\twocolumn
\lipsum[1-4]
\onecolumn
```

```
\lipsum[1-4]
\twocolumn[\centering\textbf{\large Problemas propuestos}]
\lipsum[1-4]
\end{document}
```

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula.

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula.

Problemas propuestos

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula.

La mayoría de los entornos con estrella son de dos columnas y mientras los que no tienen estrella son de una columna.

Paquete multicol

multicols

El paquete `multicol` proporciona el entorno `multicols` para obtener una salida de dos, tres o varias columnas.

```
\begin{multicols}{<number column>}
  <multicolumn text>
\end{multicols}
```

Ejemplo 4.43

```
\documentclass{article}
\usepackage{geometry}
\geometry{paperwidth=14cm,paperheight=18cm,margin=5mm}
\usepackage{lipsum}
\usepackage{multicol}
\setlength{\columnsep}{1cm}
\setlength{\columnseprule}{0.5pt}
```

```
\begin{document}
\begin{multicols}{3}
  \lipsum[1-4]
\end{multicols}
\end{document}
```

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula.

Paquete paracol

El paquete `paracol` proporciona el entorno `paracol` para obtener una salida de dos, tres o varias columnas.

Ejemplo 4.44

```
\documentclass{article}
\usepackage{geometry}
\geometry{paperwidth=14cm,paperheight=18cm,margin=5mm}
\usepackage{lipsum,amsmath}
\usepackage{paracol}
\begin{document}
\begin{paracol}{2}[\section{Introducción}]
  \lipsum[1]
```

```
  \switchcolumn
  \begin{align*}
    \sqrt{\sqrt{\sqrt{\sqrt{4}}}}
  \end{align*}
\end{paracol}

\begin{paracol}{3}[Antes de las columas divididas]
  Primera columan
  \switchcolumn
  Segunda columna
  \switchcolumn
  Tercera columna
\end{paracol}
\end{document}
```

1 Introducción

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

$$\sqrt{\sqrt{\sqrt{\sqrt{4}}}}$$

Antes de las columas divididas

Primera columan   Segunda columna   Tercera columna

Sintaxis del entorno `paracol`:

```
\begin{paracol}{<número de columnas>}[<texto>]
  <texto a la izquierda>
  \switchcolumn
  <texto a la derecha con respecto al anterior columna>
```

```
    \switchcolumn
    <texto a la derecha con respecto al anterior columna>
  \end{paracol}
```

El comando `\switchcolumn[<number>]` cambia de la columna con respecto de la columna `<number>` y si no se especifica el argumento opcional toma el anterior columna de referencia. El conteo de la columna comienza de 0.

DISEÑO DE PÁGINA

En este capítulo nos enfocaremos en estudiar los encabezados, pies de página y otros elementos que contienen una página.

5.1 Dimensiones geométricas

Las páginas de un documento LaTeX consisten en:

1. Área de texto (body or type area),
2. títulos (estos títulos se pueden colocar en la parte superior o inferior de la página) y
3. notas en el margen (margin note).

Los "títulos" como el nombre del capítulo y el número de página cambian automáticamente para cada página. Las notas en el margen (margin note) generalmente se utilizan en libros. En libros de dos caras (twoside), las notas de margen se ubican a la derecha del body cuando el número de página es impar y cuando el número de página es par se ubican a la izquierda.

Partes de documento LaTeX

En los documentos LaTeX se puede establecer si el libro o el artículo se va a imprimir anverso-reverse (twoside), también se puede establecer si las dos caras son simétricas o asimétricas (ver figura 5.1).

1. `one inch + \hoffset`
2. `one inch + \voffset`
3. `\oddsidemargin`
4. `\topmargin`
5. `\headheight`
6. `\headsep`
7. `\textheight`
8. `\textwidth`
9. `\marginparsep`
10. `\marginparwidth`
11. `\footskip`

Parámetros de la página:

`\paperheight`

`\paperwidth`

`\textheight`

1. `\paperheight`: Altura del papel.
2. `\paperwidth`: Ancho del papel.

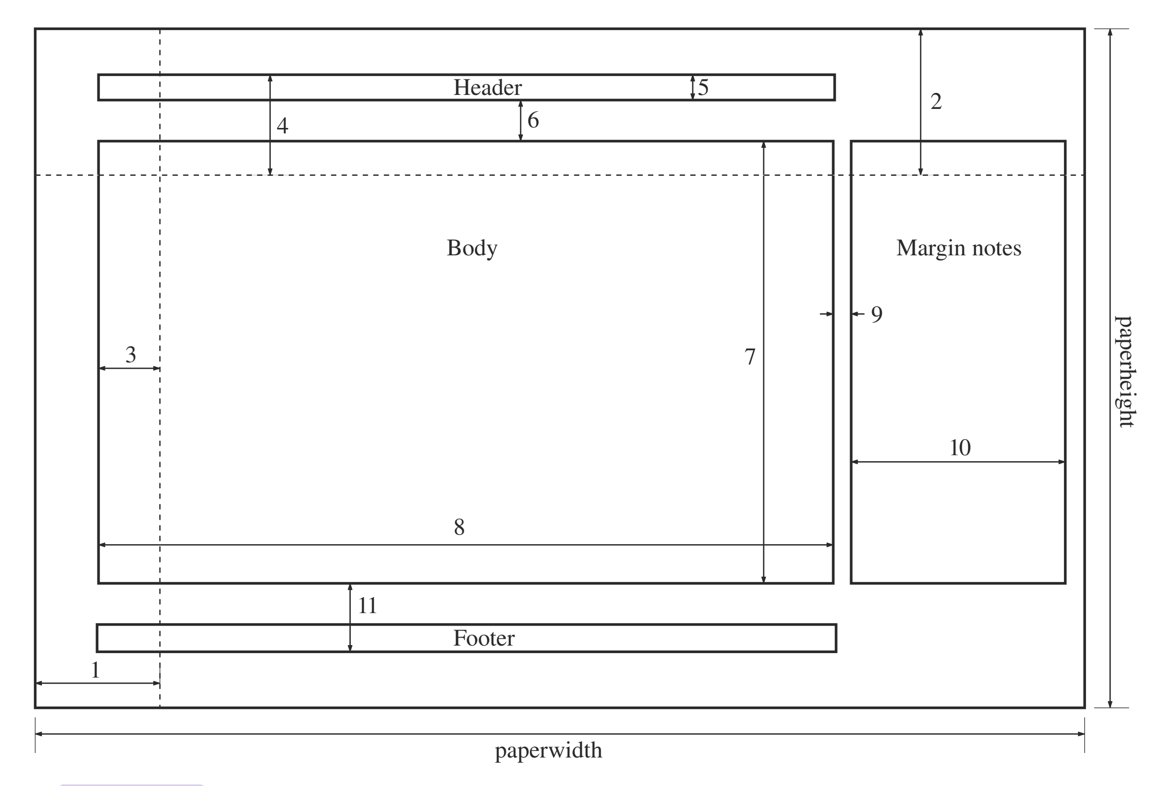


Figura 5.1

3. `\textheight`: Altura del cuerpo (sin encabezado ni pie de página).
4. `\textwidth`: Ancho del cuerpo.
5. `\columnsep`: Ancho del espacio entre columnas de texto.
6. `\columnseprule`: Ancho de la línea vertical que separa las columnas.
7. `\columnwidth`: Ancho de una sola columna en modo multicolumna.
8. `\linewidth`: Ancho de la línea de texto actual. Generalmente es igual a `\columnwidth` pero puede ser de diferente valor dentro de los entornos que cambian sus márgenes.
9. `\evensidemargin`: Es el espacio adicional que se agrega a la izquierda de las páginas pares (en documentos de dos caras).
10. `\oddsidemargin`: Es el espacio adicional que se agrega a la izquierda de las páginas impares (en documentos de dos caras). Este comando y el comando anterior desplazan la página horizontalmente.
11. `\footskip`: Distancia vertical entre las líneas base de la última línea de texto y la línea del pie de página.
12. `\headheight`: Altura del encabezado.

\textwidth
\columnsep
\columnsep␣rule
\columnwid␣th
\linewidth
\evensidem␣argin
\oddsidema␣rgin
\footskip
\headheight
\headsep

13 `\headsep`: Separación vertical entre el encabezado y el cuerpo.

14 `\topmargin`: Espacio vertical adicional parte superior del encabezado. Este comando desplaza verticalmente la página.

`\topmargin`

`\marginparpush`

`\marginparsep`

`\marginparwidth`

15 `\marginparpush`: Espacio vertical mínimo entre nostas de margen.

16 `\marginparsep`: Espacio horizontal entre notas de margen y el cuerpo.

17 `\marginparwidth`: Ancho de las notas de margen.

Nombre	Dimensión
letterpaper	81/2×11 in
legalpaper	81/2×14 in
executivepaper	71/4×101/2 in
a4paper	81/4×113/4 in ó 210×297 mm
a5paper	57/8×81/4 in ó 148×210 mm
b5paper	7×97/8 in ó 176×250 mm

Tabla 5.10

Tamaños de páginas definidos en LaTeX

Especificación de tamaño de página

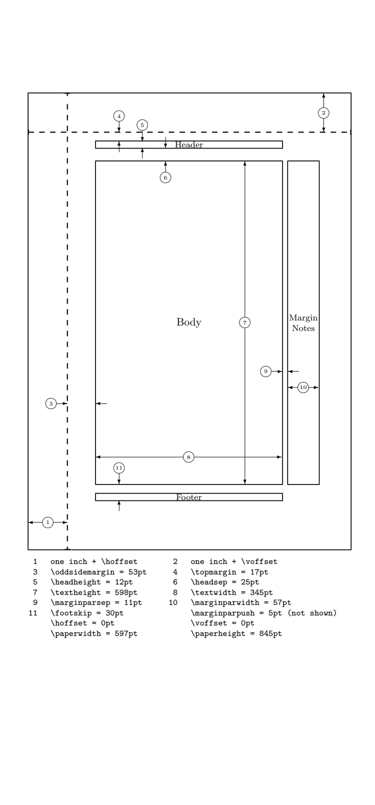
En la tabla 5.10 se muestran los tamaños de páginas definidos en LaTeX. Para utilizar algún tamaño de una página en un artículo, libro o informe, se debe especificar de la siguiente manera:

```
\documentclass[a4paper]{article}
```

En este caso estamos tomando el tamaño `a4paper`. En algunas clases existen más opciones para el argumento opcional del comando `\documentclass`.

Ejemplo 5.1

```
\documentclass[a4paper]{article}
\usepackage{layout}
\begin{document}
  \layout
\end{document}
```

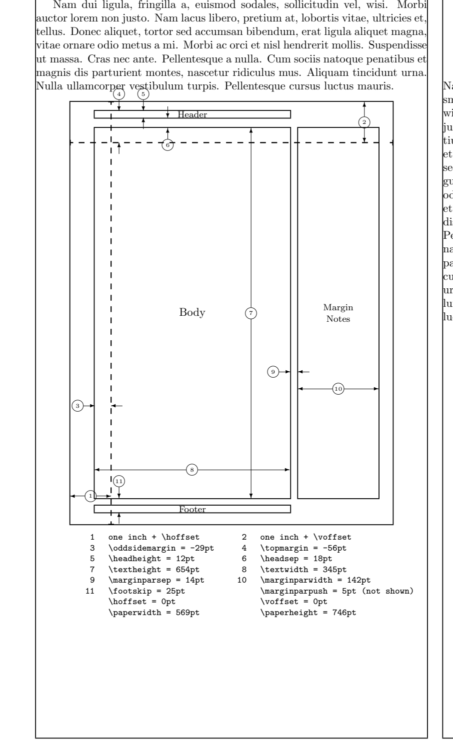


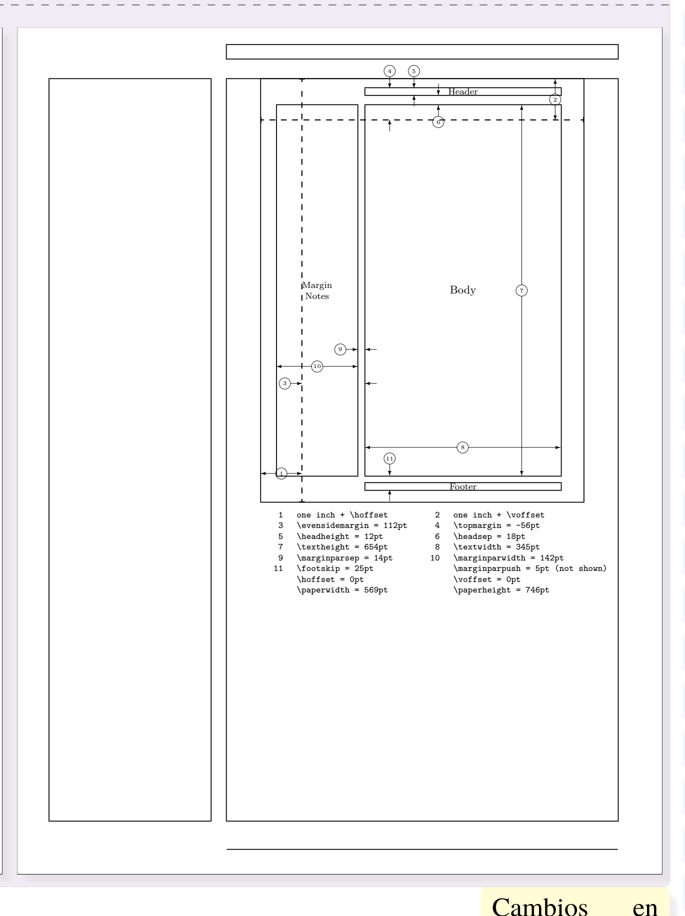
Cuando se realiza algún cambio en algún parámetro (como `paperwidth`) del diseño de página de un documento LaTeX, afecta directamente a otros parámetros del tamaño del papel; por ejemplo,

Ejemplo 5.2

```
\documentclass{book}
\paperwidth = 20cm
\paperheight = 26.25cm
\textheight = 23cm
\marginparwidth = 5cm
\marginparsep = 5mm
\topmargin = -2cm
\oddsidemargin = -1.04cm
\evensidemargin = 3.96cm
\usepackage{layout}
\usepackage{lipsum}
\usepackage{showframe}
\begin{document}
  \lipsum[2]
```

```
  \marginpar{\lipsum[2]}
  \begin{center}
    \layout
  \end{center}
\end{document}
```





Cambios en parámetros

5.2 Diseño de página

Para cambiar el diseño de una página es necesario cambiar las longitudes como `\textwidth`. Todos los parámetros de diseño de una página son longitudes; por lo tanto, se pueden incrementar, decrementar o establecer a un valor.

Definición de parámetros de una página

Los parámetros de diseño de página se definen en una clase o en un paquete, y también pueden definirse en el preámbulo. No sólo se pueden definir en el preámbulo, también es posible cambiar los valores de las longitudes en el body (cuerpo del documento LaTeX), pero esto no se recomienda debido a que el diseño se desordena. Si se realizan cambios en el body, entonces se recomienda compilar una y otra vez (en cada compilación se debe variar las longitudes) hasta obtener el resultado deseado.

`\baselineskip`

La longitud `\baselineskip` define los espacios entre las líneas de texto.

Ejemplo 5.3

```
\documentclass{book}
\usepackage{lipsum}
\begin{document}
\baselineskip = 1cm
\lipsum[2]
\end{document}
```

1

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

`\parskip` La longitud `\parskip` define los espacios entre los párrafos.

`\parindent` La longitud `\parindent` define la sangría de todos los párrafos.

`\spaceskip` La longitud `\spaceskip` define los espacios entre las palabras.

`\topskip` La longitud `\topskip` define el espacio entre el encabezado y la primera línea de texto del body.

Para los cálculos de longitudes considerar:

```
widthPaper = 1in + \oddsidemargin + \textwidth + \evensidemargin + 1in
```

Para indicar que un documento LaTeX es de 2 caras se debe utilizar la opción `twoside` como argumento opcional de clase. La variable booleana `\@twoside` imprime el valor verdadero y falsedad de twoside.

twoside

Ejemplo 5.4

```
\documentclass[twoside]{book}
\usepackage{geometry}
\geometry{paperwidth=15cm, paperheight=4cm, margin=5mm}
\usepackage{lipsum}
\usepackage{ifthen}
\begin{document}
¿Este documento \LaTeX\ es de 2 caras? \ifthenelse{\boolean{@twoside}}{Sí}{No}
\end{document}
```

¿Este documento LaTeX es de 2 caras? Sí

hoffset y voffset

En esta sección estudiaremos cómo TeX establece el sistema de coordenadas y cómo se pueden modificar los parámetros para ajustar la posición y el tamaño del área de texto en un documento ya los parámetros `\hoffset` y `\voffset` suele generar confusión.

Cuando TeX comienza un nuevo documento, la intersección de las dos líneas principales (eje x y eje y) se define como el origen del sistema de coordenadas. El eje x se extiende hacia la derecha, mientras que el eje y va hacia abajo. En el caso de PostScript, este origen se encuentra en la esquina inferior izquierda de la página.

Si tienes problemas con una impresora que desplaza las páginas impresas debido a un problema de hardware, puedes ajustar los parámetros `\hoffset` y `\voffset`. Estos parámetros representan el desplazamiento horizontal y vertical, respectivamente, y están relacionados con las líneas punteadas que se describen en el texto. TeX establece un desplazamiento fijo de una pulgada desde la izquierda y una pulgada desde la parte superior de la página para los ejes.

Para modificar la posición del bloque de texto en la página, puedes cambiar los valores de `\hoffset` y `\voffset` (ambos con valor predeterminado cero). Al calcular la posición de los márgenes izquierdo y superior del bloque de texto, debes tener en cuenta este desplazamiento fijo.

Para una página impar (o cualquier página en estilo "oneside"), el bloque de texto se coloca a:

```
1in + \hoffset + \oddsidemargin
```

desde el borde izquierdo de la página. Para las páginas pares, se utiliza `\evensidemargin` en lugar de `\oddsidemargin`.

Del mismo modo, la parte superior del bloque de texto se encuentra a:

```
1in + \voffset + \topmargin + \headheight + \headsep
```

y la primera línea de base será a:

```
1in + \voffset + \topmargin + \headheight + \headsep + \topskip
```

desde el borde superior de la página.

Aunque puedes cambiar los valores de `\hoffset` y `\voffset` para ajustar la posición del bloque de texto, no se recomienda hacerlo a menos que sea necesario debido a problemas con la interpretación de las instrucciones por parte del controlador de la impresora (algo que no ocurre con frecuencia en la actualidad).

Los parámetros `\textwidth` y `\textheight` determinan el área del bloque de texto. No es posible especificar directamente la distancia desde el borde derecho o inferior del bloque de texto hasta el borde de página correspondiente.

`\hoffset` y `\voffset` se remontan a los días del TeX simple. Donald Knuth, creador de TEX, menciona brevemente estos parámetros en el TeXbook:

1. p. 251: Si deseas cambiar la posición de la salida impresa, puedes ajustar los valores de `\hoffset` y `\voffset`. Por ejemplo, al establecer `\hoffset` en 0.5 pulgadas y `\voffset` en 1.5 pulgadas, la salida se moverá media pulgada a la derecha y 1.5 pulgadas hacia abajo de su posición normal.
2. p. 274: `\hoffset` y `\voffset` son parámetros de dimensión que afectan el desplazamiento horizontal y vertical en la salida de `\shipout` (cuando TeX envía una página al archivo de salida).
3. p. 342: Knuth proporciona ejemplos de cómo cambiar los valores de los parámetros importantes que no sean `\hsize` y `\vsize`. Por ejemplo, al establecer `\hoffset` en 1.5 pulgadas, toda la salida se desplazará a la derecha en una pulgada y media.

Visualización de diseño de páginas

Visualización de diseño de páginas

Para visualizar el diseño de página utilizaremos dos paquetes: `layout` y `layouts`. También podemos utilizar el paquete `showframe` o habilitar la opción `showframe` del paquete `geometry`. Anteriormente hemos utilizado el paquete `layout` para visualizar el diseño de las páginas. El paquete `layouts` tiene más opciones, si desea experimentar con estas opciones puede revisar la documentación oficial `https://linorg.usp.br/CTAN/macros/latex/contrib/layouts/layman.pdf`.

Paquete geometry

Geometría de una página

En la figura 5.3 se muestran las longitudes de diseño de página de un documento LATEX. El diseño mostrado

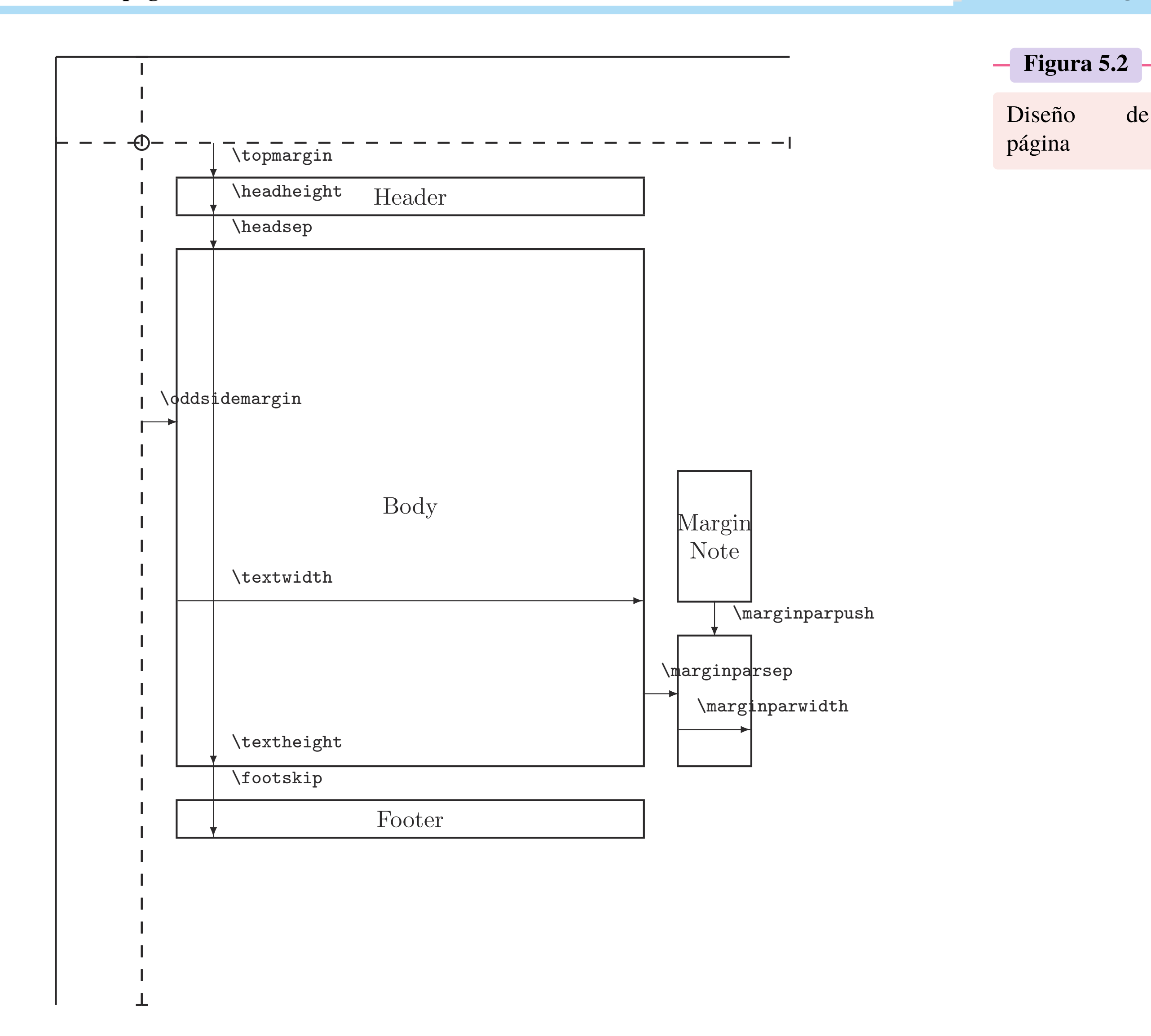


Figura 5.2
Diseño de página

en la figura 5.3 contiene el área imprimible (total body) y los márgenes. En el área imprimible (total body) se tiene el body (área de texto), el header (encabezado), el footer (pie de página) y las notas en el margen. En los márgenes se tienen: el izquierdo, el derecho, el superior y el inferior. En resumen,

$$\text{paper} : \text{total body and margins}$$
$$\text{total body} : \text{body, header, footer and marginpar}$$
$$\text{margins} : \text{left, right, top and bottom}$$

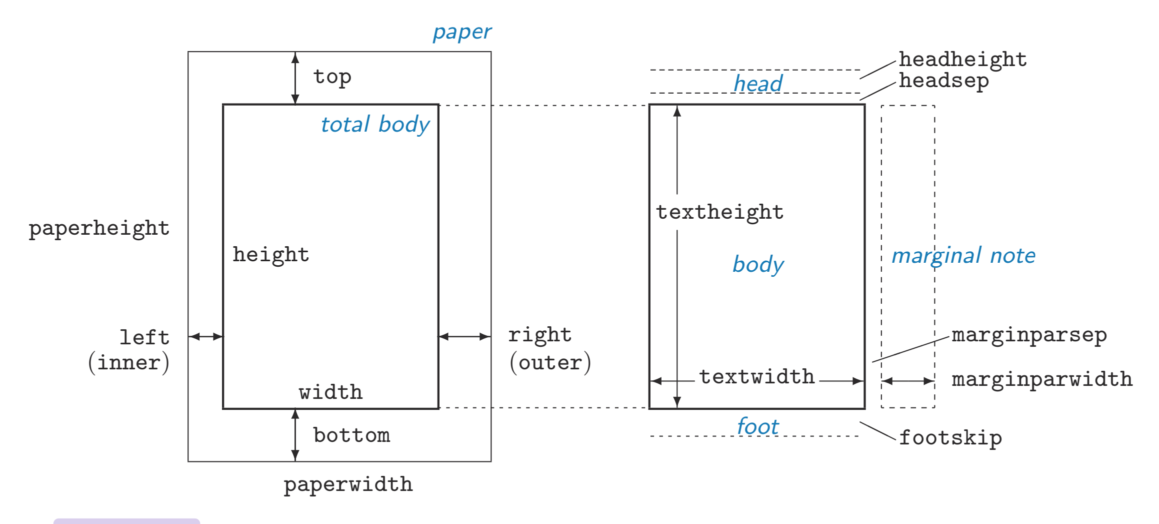
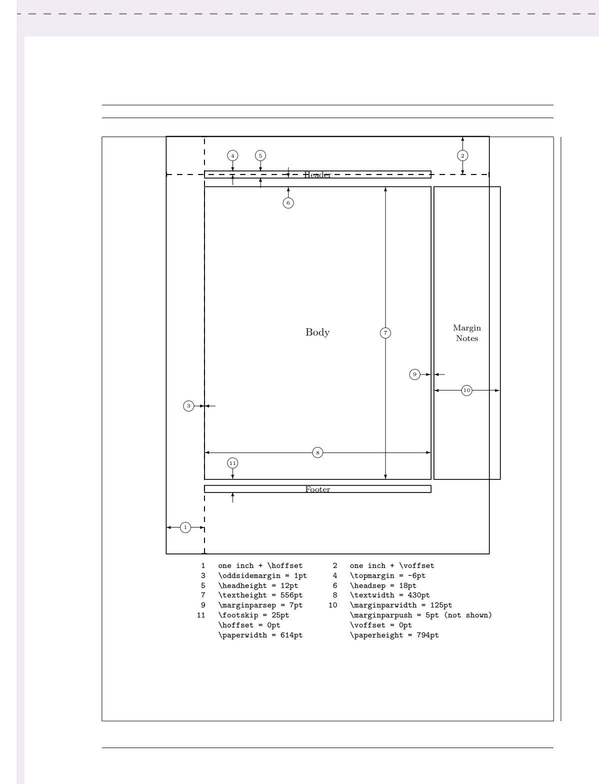


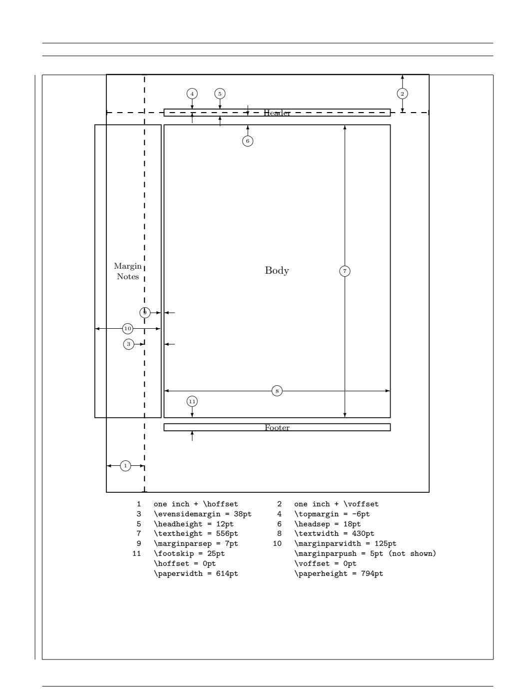
Figura 5.3

Geometría de una página

Ejemplo 5.5

```
\documentclass{book}
\usepackage{layout}
\usepackage{geometry}
\geometry{showframe}
\begin{document}
  \begin{center}
    \layout
  \end{center}
\end{document}
```





Las longitudes `left`, `right`, `top` y `bottom` son independientes de las longitudes `leftmargin` y `topmargin`. El tamaño de un papel está definido de la siguente manera:

left, right, top, bottom, leftmargin y topmargin

$$\text{paperwidth} = \text{left} + \text{width} + \text{right}$$

$$\text{paperheight} = \text{top} + \text{height} + \text{bottom}$$

Cuando se utiliza las opciones `includehead` y `includefoot` se incluyen el encabezado (header) y el pie de página (footer) en total body (ver figura 5.4).

includehead y includefoot

```
\usepackage{geometry}
\geometry{includehead, includefoot}
```

Uso del paquete geometry

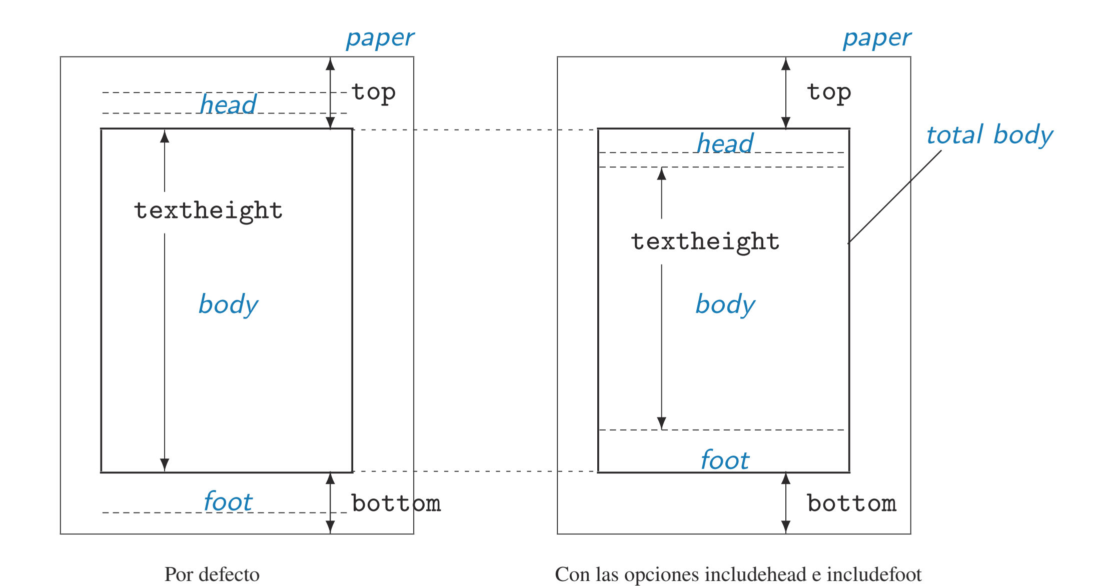
Para utilizar el paquete geometry se debe utilizar el comando `\usepackage` de LaTeX.

geometry

```
\usepackage{geometry}
```

El paquete geometry proporciona los siguientes comandos:

Figura 5.4 Geometría de una página con las opciones includehead e includefoot



1. \geometry{<opciones de geometria>}

2. <opciones>

3. \restoregeometry

4. \savegeometry{<nombre de geometria>}

5. \loadgeometry{<nombre de geometria>}

\geometry

El comando \geometry define las longitudes de diseño de página.

Ejemplo 5.6

```
\documentclass{book}
\usepackage{geometry}
\geometry{paperwidth=20cm, paperheight=26cm, left=1.5cm, right=1cm, top=1cm, bottom=1cm}
\usepackage{lipsum}
\begin{document}
  \lipsum[1-5]
\end{document}
```

1

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula.

Fusce mauris. Vestibulum luctus nibh at lectus. Sed bibendum, nulla a faucibus semper, leo velit ultricies tellus, ac venenatis arcu wisi vel nisl. Vestibulum diam. Aliquam pellentesque, augue quis sagittis posuere, turpis lacus congue quam, in hendrerit risus eros eget felis. Maecenas eget erat in sapien mattis porttitor. Vestibulum porttitor. Nulla facilisi. Sed a turpis eu lacus commodo facilisis. Morbi fringilla, wisi in dignissim interdum, justo lectus sagittis dui, et vehicula libero dui cursus dui. Mauris tempor ligula sed lacus. Duis cursus enim ut augue. Cras ac magna. Cras nulla. Nulla egestas. Curabitur a leo. Quisque egestas wisi eget nunc. Nam feugiat lacus vel est. Curabitur consectetuer.

El comando `\newgeometry` define una nueva geometría (se aplica cuando se tiene un libro o artículo con diferentes diseños de página). El comando `\newgeometry` crea un nuevo diseño de página, que se guarda con el comando `\savegeometry` y luego se puede cargar o utilizar este diseño con el comando `\loadgeometry` , y finalmente, para restaurar el diseño de página original se debe utilizar el comando `\restoregeometry` .

`\newgeometry`

`\savegeometry`

Ejemplo 5.7

```
\documentclass{book}
\usepackage{geometry}
\geometry{%
  includehead, includefoot, includemp,
  paperwidth=20cm,left=1.5cm,right=1cm, marginparsep=5mm, marginparwidth=5cm,
  paperheight=26.25cm,top=1cm,bottom=1cm,
  % showframe%
}
\usepackage{lipsum}
\begin{document}
  \chapter{Capítulo 1}
  \marginpar{\lipsum[2]}
```

```
  \lipsum[1-3]
  \newgeometry{%
    includehead, includefoot,
    left=1.5cm,right=1cm,
    top=1cm,bottom=1cm,%
  }
  \savegeometry{a}
  \loadgeometry{a}
  \twocolumn[%
    \begin{center}
      {\bfseries  PROBLEMAS RESUELTOS}
    \end{center}%
  ]
  \lipsum[1-8]
  \restoregeometry
  \lipsum[1-5]
  \marginpar{\lipsum[2]}
\end{document}
```

Chapter 1

Capítulo 1

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

1

2 *CHAPTER 1. CAPÍTULO 1*

PROBLEMAS RESUELTOS

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula.

Fusce mauris. Vestibulum luctus nibh at lectus. Sed bibendum, nulla a faucibus semper, leo velit ultricies tellus, ac venenatis arcu wisi vel nisl. Vestibulum diam. Aliquam pellentesque, augue quis sagittis posuere, turpis lacus congue quam, in hendrerit risus eros eget felis. Maecenas eget erat in sapien mattis porttitor. Vestibulum porttitor. Nulla facilisi. Sed a turpis eu lacus commodo facilisis. Morbi fringilla, wisi in dignissim interdum, justo lectus sagittis dui, et vehicula libero dui cursus dui. Mauris tempor ligula sed lacus. Duis cursus enim ut augue. Cras ac magna. Cras nulla. Nulla egestas. Curabitur a leo. Quisque egestas wisi eget nunc. Nam feugiat lacus vel est. Curabitur consectetuer.

Suspendisse vel felis. Ut lorem lorem, interdum eu, tincidunt sit amet, laoreet vitae, arcu. Aenean faucibus pede eu ante. Praesent enim elit, rutrum at, molestie non, nonummy vel, nisl. Ut lectus eros, malesuada sit amet, fermentum eu, sodales cursus, magna. Donec eu purus. Quisque vehicula, urna sed ultricies auctor, pede lorem egestas dui, et convallis elit erat sed nulla. Donec luctus. Curabitur et nunc. Aliquam dolor odio, commodo pretium, ultricies non, pharetra in, velit. Integer arcu est, nonummy in, fermentum faucibus, egestas vel, odio.

Sed commodo posuere pede. Mauris ut est. Ut quis purus. Sed ac odio. Sed vehicula hendrerit sem. Duis non odio. Morbi ut dui. Sed accumsan risus eget odio. In hac habitasse platea dictumst. Pellentesque non elit. Fusce sed justo eu urna porta tincidunt. Mauris felis odio, sollicitudin sed, volutpat a, ornare ac, erat. Morbi quis dolor. Donec pellentesque, erat ac sagittis semper, nunc dui lobortis purus, quis congue purus metus ultricies tellus. Proin et quam. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos hymenaeos. Praesent sapien turpis, fermentum vel, eleifend faucibus, vehicula eu, lacus.

Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Donec odio elit, dictum in, hendrerit sit amet, egestas sed, leo. Praesent feugiat sapien aliquet odio. Integer vitae justo. Aliquam vestibulum fringilla lorem. Sed neque lectus, consectetuer at, consectetuer sed, eleifend ac, lectus. Nulla facilisi. Pellentesque eget lectus. Proin eu metus. Sed porttitor. In hac habitasse platea dictumst. Suspendisse eu lectus. Ut mi mi, lacinia sit amet, placerat et, mollis vitae, dui. Sed ante tellus, tristique ut, iaculis eu, malesuada ac, dui. Mauris nibh leo, facilisis non, adipiscing quis, ultrices a, dui.

3

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula.

Fusce mauris. Vestibulum luctus nibh at lectus. Sed bibendum, nulla a faucibus semper, leo velit ultricies tellus, ac venenatis arcu wisi vel nisl. Vestibulum diam. Aliquam pellentesque, augue quis sagittis posuere, turpis lacus congue quam, in hendrerit risus eros eget felis. Maecenas eget erat in sapien mattis porttitor. Vestibulum porttitor. Nulla facilisi. Sed a turpis eu lacus commodo facilisis. Morbi fringilla, wisi in dignissim interdum, justo lectus sagittis dui, et vehicula libero dui cursus dui. Mauris tempor ligula sed lacus. Duis cursus enim ut augue. Cras ac magna. Cras nulla. Nulla egestas. Curabitur a leo. Quisque egestas wisi eget nunc. Nam feugiat lacus vel est. Curabitur consectetuer.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis

`\loadgeometry`

`\restoregeometry`

Tipos de opciones

El paquete `geometry` utiliza opciones de tipo `key=value`. Existen los valores de tipo booleano, valores simples, dobles y triples.

1. Boleano:

```
<key> = true|false
```

```
showframe = true
```

2. Simple:

```
<key> = <valor>
```

```
paperwidth = 20cm
```

3. Doble:

```
<key> = {<valor1>, <valor2>}
```

```
body = {10cm, 20cm}
```

4 Triple:

```
<key> = {<valor1>, <valor2>, <valor3>}

hdivide = {2cm, *, 1cm}
```

Layout size

Opciones de tamaño del diseño de página (layout size) se muestran en la figura 5.5.

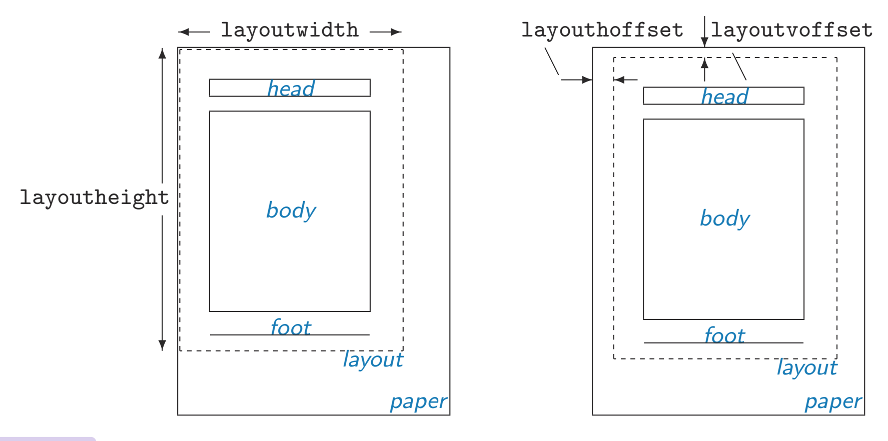


Figura 5.5

```
layout = <paper name>
layoutwidth = <length>
layoutheight = <length>
layoutsize = {<width, height>}
layoutsize = <length>
layouthoffset = <length>
layoutvoffset = <length>
layoutoffset = {<hoffset>, <voffset>}
```

Body size

Opciones de tamaño total del cuerpo (total body):

1. `hscale`: Escala el cuerpo total con el ancho del papel; por ejemplo, `hscale = 0.8` es equivalente a `width=0.8\paperwidth`.

2. `vscale`: Escala el cuerpo total con el alto del papel; por ejemplo, `vscale = 0.8` es equivalente a `height=0.8\paperheight`.

3. `scale`: Es de sintaxis `scale = {<h-scale>, <v-scale>}` y se analiza de manera similar que los anteriores.

4. width|totalwidth: Define el ancho del cuerpo total y es del mismo valor que \textwidth, pero si se habilita la opción includemp, entonces, width⩾textwidth. Se debe tener cuidado en especificar al mismo tiempo el textwidth y width.
5. height|totalheight: Define el alto del cuerpo total (excluye pie de página y encabezado, por default) y es del mismo valor que \textheight, pero si se habilita la opción includefoot o includehead, entonces, se incluye foot y head en la longitud height, como de la misma forma está incluido textheight.
6. total: Es de sintaxis total = {<width>, <height>} y define el ancho y alto del cuerpo total.
7. textwidth: Especifica el ancho del body (área de texto).
8. textheight: Especifica el alto del body (área de texto).
9. text|body: Es de sintaxis text = <lenght> y body = {<width>, <height>}.
10. includehead: Incluye el encabezado en en total body. Por default, no se incluye.
11. includefoot: Incluye el pie de página en total body. Por default, no se incluye.
12. includeheadfoot: Realiza las dos operaciones anteriores.
13. includemp: Incluye el margen de página en total body. Por default, no se incluye.
14. includeall: Es lo mismo que habilitar las opciones includeheadfoot y includemp.
15. ignorehead: Ignora el encabezado y su longitud.
16. ignorefoot: Ignora el pie de página y su longitud.
17. ignoreheadfoot: Ignora el encabezado y el pie de página y sus respectivos longitudes.
18. ignoremp.
19. ignoreall.

Margin size

Opciones de tamaño del margen:

1. left.
2. right.
3. top.
4. bottom.
5. twosite: Los márgenes izquierdo y derecho se intercambian en el reverso de la página.
6. asymmetric: Los márgenes no se intercambian.

Paquete crop

Este paquete sirve para generar marcas de recorte.

5.3 Numeraciones de páginas y marcadores

Números de página

\thepage

\pagenumbering

Los números de páginas se controlan por el contador page y para obtener su representación se obtiene con \thepage. El comando \thepage no está definido en nucleo de LaTeX, si no que surge cuando se ejecuta el comando \pagenumbering por primera vez.

El comando \pagenumbering se ejecuta en una clase.

Para las referencias cruzadas de páginas se utiliza el comando \pageref.

Ejemplo 5.8

```
\documentclass{article}
\usepackage{geometry}
\geometry{paperwidth=10cm, paperheight=5cm, margin=5mm}
\begin{document}
  \label{p1} La página es \pageref{p1}.

  La página actual es \thepage.
\end{document}
```

La página es 1.
La página actual es 1.

\pagenumbering

Al momento de escribir libros, generalmente las primeras páginas se enumeran en romano y luego se resetea a 1 y se continúa con la numeración de páginas con números arábigos. Para realizar estos cambios existe el comando

```
\pagenumbering{<style>}
```

Este comando hace que el contador de páginas sea 1 y redefine \thepage a \layout{page}

Ejemplo 5.9

```
\documentclass{book}
\usepackage{geometry}
\geometry{paperwidth=15cm, paperheight=20cm}
\usepackage{lipsum}
\begin{document}
  \pagenumbering{Roman}
  \author{Heber}
  \title{Libro de prueba}
  \maketitle
  \pagenumbering{arabic}
  \chapter{Capitulo 1}
  \lipsum[1-4]
\end{document}
```

Libro de prueba

Heber

June 4, 2023

Chapter 1

Capitulo 1

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec

1

2 CHAPTER 1. CAPITULO 1

ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula.

Ejemplo 5.10

```
\documentclass{book}
\usepackage{geometry}
\geometry{paperwidth=15cm, paperheight=20cm}
\usepackage{lipsum}
\begin{document}
  \frontmatter
  \author{Heber}
  \title{Libro de prueba}
  \maketitle
  \mainmatter
  \chapter{Capitulo 1}
  \lipsum[1-4]
\end{document}
```

Libro de prueba

Heber

June 4, 2023

ii

2 CHAPTER 1. CAPITULO 1

Chapter 1

Capitulo 1

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec

1

ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula.

5.4 Estilos de página

`\pagestyle`

`\thispagestyle`

En LaTeX se pueden crear diferentes estilos de página y para seleccionar o utilizar un estilo creado se debe recurrir al comando `\pagestyle` o `\thispagestyle`. El primer comando afecta a las demás páginas y mientras tanto el segundo comando solo afecta a una página determinada (actual), ambos comandos tienen un argumento obligatorio. Generalmente este comando se utiliza cuando se escriben libros debido a que un libro puede contener diferentes estilos de página.

estilo de página `empty`

Si se desea que una página o páginas no contengan la numeración de páginas y el encabezado esté vacío, se puede utilizar el comando `\thispagestyle{empty}` o `\pagestyle{empty}`, como sucede en las portadas de libros. También existen otros estilos de páginas, como :

1. `empty` (el encabezado y el pie de página están vacíos),
2. `plain` (el encabezado está vacío y, mientras tanto, existe la numeración de páginas),
3. `headings` (el encabezado tiene información proporcionada por la clase y el pie de página está vacío) y
4. `myheadings` (es similar al anterior, solo que en este caso el usuario especifica el encabezado superior).

Un diseño de página, como plain, podemos redefinir de la siguiente forma:

```
\let\ps@plain=\ps@empty
```

Los comandos \markboth y \markright generan los encabezados.

Los comandos con estrella como \chapter no llama al \chaptermark si no que llama a @mkboth.

\markboth y \markright

Estilos de páginas de bajo nivel

El kernel de LaTeX tiene cuatro comandos para personalizar el encabezado:

1. \@oddhead
2. \@oddfoot
3. \@evenhead
4. \@evenfoot

Paquete fancyhdr

Uso del paquete fancyhdr

Sintaxis para utilizar este paquete:

fancyhdr

```
\usepackage{fancyhdr}
```

Este paquete nos ayuda a personalizar los encabezados y los pies de página. Para indicar que se va a utilizar el estilo de página proporcionado por el paquete fancyhdr, se debe invocar el comando

\pagestyle {fancy}

```
\pagestyle{fancy}
```

El diseño de página fancy se muestra en la figura 5.6.

LeftHeader	CenteredHeader	RightHeader
	page body	
LeftFooter	CenteredFooter	RightFooter

Figura 5.6

Ejemplo 5.11

```
\documentclass{article}
\usepackage{lipsum}
\usepackage{fancyhdr}
\fancyhead[L]{Left header}
```

```
\fancyhead[C]{Center header}
\fancyhead[R]{Right header}
\fancyfoot[L]{Left footer}
\fancyfoot[C]{Center footer}
\fancyfoot[R]{Right footer}
\renewcommand{\headrulewidth}{0.4pt}
\renewcommand{\footrulewidth}{2pt}
\begin{document}
  \pagestyle{fancy}
  \section{Sección 1}
  \lipsum[1-7]
\end{document}
```

Left header Center header Right header

1 Sección 1

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula.

Fusce mauris. Vestibulum luctus nibh at lectus. Sed bibendum, nulla a faucibus semper, leo velit ultricies tellus, ac venenatis arcu wisi vel nisl. Vestibulum diam. Aliquam pellentesque, augue quis sagittis posuere, turpis lacus congue quam, in hendrerit risus eros eget felis. Maecenas eget erat in sapien mattis porttitor. Vestibulum porttitor. Nulla facilisi. Sed a turpis eu lacus commodo facilisis. Morbi fringilla, wisi in dignissim interdum, justo lectus sagittis dui, et vehicula libero dui cursus dui. Mauris tempor ligula sed lacus. Duis cursus enim ut augue. Cras ac magna. Cras nulla. Nulla egestas. Curabitur a leo. Quisque

Left footer Center footer Right footer

Left header Center header Right header

egestas wisi eget nunc. Nam feugiat lacus vel est. Curabitur consectetuer.

Suspendisse vel felis. Ut lorem lorem, interdum eu, tincidunt sit amet, laoreet vitae, arcu. Aenean faucibus pede eu ante. Praesent enim elit, rutrum at, molestie non, nonummy vel, nisl. Ut lectus eros, malesuada sit amet, fermentum eu, sodales cursus, magna. Donec eu purus. Quisque vehicula, urna sed ultricies auctor, pede lorem egestas dui, et convallis elit erat sed nulla. Donec luctus. Curabitur et nunc. Aliquam dolor odio, commodo pretium, ultricies non, pharetra in, velit. Integer arcu est, nonummy in, fermentum faucibus, egestas vel, odio.

Sed commodo posuere pede. Mauris ut est. Ut quis purus. Sed ac odio. Sed vehicula hendrerit sem. Duis non odio. Morbi ut dui. Sed accumsan risus eget odio. In hac habitasse platea dictumst. Pellentesque non elit. Fusce sed justo eu urna porta tincidunt. Mauris felis odio, sollicitudin sed, volutpat a, ornare ac, erat. Morbi quis dolor. Donec pellentesque, erat ac sagittis semper, nunc dui lobortis purus, quis congue purus metus ultricies tellus. Proin et quam. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos hymenaeos. Praesent sapien turpis, fermentum vel, eleifend faucibus, vehicula eu, lacus.

Left footer Center footer Right footer

Para libros de doble cara, se debe especificar la información que se va a imprimir en las páginas pares (even) e impares (odd). Para imprimir la cadena “center” en el encabezado de las páginas impares (odd), se debe añadir O después de la posición, en este caso C, y para pares (even), se debe añadir E:

Ejemplo 5.12

```
\documentclass{book}
\usepackage[x11names,table,svgnames,usenames,dvipsnames]{xcolor}
\usepackage{lipsum}
\usepackage{fancyhdr}
\fancyhead[RO,RE]{\color{Blue}Right header}
\fancyhead[COE]{\color{red}Heber}
\fancyhead[LO,LE]{\color{magenta}Left header}
\fancyfoot[CO,CE]{\bfseries\color{Salmon}\thepage}
\renewcommand{\headrulewidth}{0.4pt}
\renewcommand{\footrulewidth}{2pt}
\begin{document}
  \pagestyle{fancy}
  \lipsum[1-8]
\end{document}
```

Left header Heber Right header

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula.

Fusce mauris. Vestibulum luctus nibh at lectus. Sed bibendum, nulla a faucibus semper, leo velit ultricies tellus, ac venenatis arcu wisi vel nisl. Vestibulum diam. Aliquam pellentesque, augue quis sagittis posuere, turpis lacus congue quam, in hendrerit risus eros eget felis. Maecenas eget erat in sapien mattis porttitor. Vestibulum porttitor. Nulla facilisi. Sed a turpis eu lacus commodo facilisis. Morbi fringilla, wisi in dignissim interdum, justo lectus sagittis dui, et vehicula libero dui cursus dui. Mauris tempor ligula sed lacus. Duis cursus enim ut augue. Cras ac magna. Cras nulla. Nulla egestas. Curabitur a leo. Quisque egestas wisi eget nunc. Nam feugiat lacus vel est. Curabitur consectetuer.

Suspendisse vel felis. Ut lorem lorem, interdum eu, tincidunt sit amet, lao-

1

Left header Heber Right header

reet vitae, arcu. Aenean faucibus pede eu ante. Praesent enim elit, rutrum at, molestie non, nonummy vel, nisl. Ut lectus eros, malesuada sit amet, fermentum eu, sodales cursus, magna. Donec eu purus. Quisque vehicula, urna sed ultricies auctor, pede lorem egestas dui, et convallis elit erat sed nulla. Donec luctus. Curabitur et nunc. Aliquam dolor odio, commodo pretium, ultricies non, pharetra in, velit. Integer arcu est, nonummy in, fermentum faucibus, egestas vel, odio.

Sed commodo posuere pede. Mauris ut est. Ut quis purus. Sed ac odio. Sed vehicula hendrerit sem. Duis non odio. Morbi ut dui. Sed accumsan risus eget odio. In hac habitasse platea dictumst. Pellentesque non elit. Fusce sed justo eu urna porta tincidunt. Mauris felis odio, sollicitudin sed, volutpat a, ornare ac, erat. Morbi quis dolor. Donec pellentesque, erat ac sagittis semper, nunc dui lobortis purus, quis congue purus metus ultricies tellus. Proin et quam. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos hymenaeos. Praesent sapien turpis, fermentum vel, eleifend faucibus, vehicula eu, lacus.

Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Donec odio elit, dictum in, hendrerit sit amet, egestas sed, leo. Praesent feugiat sapien aliquet odio. Integer vitae justo. Aliquam vestibulum fringilla lorem. Sed neque lectus, consectetuer at, consectetuer sed, eleifend ac, lectus. Nulla facilisi. Pellentesque eget lectus. Proin eu metus. Sed porttitor. In hac habitasse platea dictumst. Suspendisse eu lectus. Ut mi mi, lacinia sit amet, placerat et, mollis vitae, dui. Sed ante tellus, tristique ut, iaculis eu, malesuada ac, dui. Mauris nibh leo, facilisis non, adipiscing quis, ultrices a, dui.

2

En general, para personalizar el encabezado y pie de página existen los comandos:

\fancyhead

\fancyfoot

\fancyhf

```
\fancyhead[<places>]{<fields>}
\fancyfoot[<places>]{<fields>}
\fancyhf[<places>]{<fields>}
```

Los <places> se pueden combinar de acuerdo de acuerdo con la tabla 5.11.

E	Even page
O	Odd page
L	Left field
C	Center field
R	Right field
H	Header
F	Footer

Tabla 5.11

Posiciones fancyhdr

Macros fancyhdr

Macros proporcionados por el paquete `fancyhdr`:

1. `\headrulewidth`
2. `\footrulewidth`. Este macro y el anterior determinan el grosor de la línea.
3. `\headruleskip`
4. `\footruleskip`. Este macro y el anterior determinan la distancia entre la línea y el encabezado o pie de página.
5. `\headrule`
6. `\footrule`. Este macro y el anterior son utilizados para redefinir el estilo de la línea.
7. `\fancyheadinit`.
8. `\fancyfootinit`. Este macro y el anterior son utilizados para definir código de inicialización.
9. `\fancyhfinit`. Es lo mismo que los dos anteriores.
10. `\fancycenter`.
11. `\iftopfloat`.
12. `\ifbotfloat`.
13. `\iffloatpage`.
14. `\iffootnote`.
15. `\fancypagestyle`. Este comando es de sintaxis:

```
\fancypagestyle{<style name>}[<base style>]{<definitions>}
```

Este comando es utilizado para redefinir el estilo de página.

Redefiniendo estilo de página plain

Cuando se invoca el comando `chapter` automáticamente esa página utiliza el estilo de página `plain`.

Ejemplo 5.13

```
\documentclass{book}
\usepackage[x11names,table,svgnames,usenames,dvipsnames]{xcolor}
\usepackage{tikz}
\usepackage{lipsum}
\usepackage{fancyhdr}
\fancypagestyle{plain}{%
  \fancyhf{}%deja en blanco el encabezado y el pie de página
  \fancyfoot[C]{\tikz{\node[rectangle, rounded corners=2pt, fill=Salmon!50!white]{\bfseries\thepage};}}
  \renewcommand{\headrulewidth}{0.4pt}
  \renewcommand{\footrulewidth}{2pt}%
}
\begin{document}
  \chapter{Cap 1}
  \lipsum[1-8]
\end{document}
```

Chapter 1

Cap 1

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. In

1

2 CHAPTER 1. CAP 1

hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula.

Fusce mauris. Vestibulum luctus nibh at lectus. Sed bibendum, nulla a faucibus semper, leo velit ultricies tellus, ac venenatis arcu wisi vel nisl. Vestibulum diam. Aliquam pellentesque, augue quis sagittis posuere, turpis lacus congue quam, in hendrerit risus eros eget felis. Maecenas eget erat in sapien mattis porttitor. Vestibulum porttitor. Nulla facilisi. Sed a turpis eu lacus commodo facilisis. Morbi fringilla, wisi in dignissim interdum, justo lectus sagittis dui, et vehicula libero dui cursus dui. Mauris tempor ligula sed lacus. Duis cursus enim ut augue. Cras ac magna. Cras nulla. Nulla egestas. Curabitur a leo. Quisque egestas wisi eget nunc. Nam feugiat lacus vel est. Curabitur consectetuer.

Suspendisse vel felis. Ut lorem lorem, interdum eu, tincidunt sit amet, laoreet vitae, arcu. Aenean faucibus pede eu ante. Praesent enim elit, rutrum at, molestie non, nonummy vel, nisl. Ut lectus eros, malesuada sit amet, fermentum eu, sodales cursus, magna. Donec eu purus. Quisque vehicula, urna sed ultricies auctor, pede lorem egestas dui, et convallis elit erat sed nulla. Donec luctus. Curabitur et nunc. Aliquam dolor odio, commodo pretium, ultricies non, pharetra in, velit. Integer arcu est, nonummy in, fermentum faucibus, egestas vel, odio.

Sed commodo posuere pede. Mauris ut est. Ut quis purus. Sed ac odio. Sed vehicula hendrerit sem. Duis non odio. Morbi ut dui. Sed accumsan risus eget odio. In hac habitasse platea dictumst. Pellentesque non elit. Fusce sed justo eu urna porta tincidunt. Mauris felis odio, sollicitudin sed, volutpat a, ornare ac, erat. Morbi quis dolor. Donec pellentesque, erat ac sagittis semper, nunc dui lobortis purus, quis congue purus metus ultricies tellus. Proin et quam. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos hymenaeos. Praesent sapien turpis, fermentum vel, eleifend faucibus, vehicula eu, lacus.

Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Donec odio elit, dictum in, hendrerit sit amet, egestas sed, leo. Praesent feugiat sapien aliquet odio. Integer vitae justo. Aliquam vestibulum fringilla lorem. Sed neque lectus, consectetuer at, consectetuer sed, eleifend ac, lectus. Nulla facilisi. Pellentesque eget lectus. Proin eu metus. Sed porttitor. In hac habitasse platea dictumst. Suspendisse eu lectus. Ut mi mi, lacinia sit amet, placerat et, mollis vitae, dui. Sed ante tellus, tristique ut, iaculis eu, malesuada ac, dui. Mauris nibh leo, facilisis non, adipiscing quis, ultrices a, dui.

Definiendo nuevos estilos de página

Ejemplo 5.14

```
\documentclass{book}
\usepackage[x11names,table,svgnames,usenames,dvipsnames]{xcolor}
\usepackage{tikz}
\usepackage{lipsum}
\usepackage{fancyhdr}
\fancypagestyle{myestilo}{%
  \fancyhf{}%deja en blanco el encabezado y el pie de página
  \fancyhead[LE,RO]{\rightmark}
  \fancyhead[LO,RE]{\leftmark}
  \fancyfoot[C]{\tikz{\node[rectangle, rounded corners=2pt, fill=Salmon!50!white]{\bfseries\thepage};}}
  \renewcommand{\headrulewidth}{0.4pt}
  \renewcommand{\footrulewidth}{2pt}%
}
\begin{document}
  \chapter{Capítulo de prueba}
  \section{Sección de prueba}
```

```
  \lipsum[2]
  \newpage
  \pagestyle{myestilo}
  \lipsum[1-4]
\end{document}
```

Chapter 1

Capítulo de prueba

1.1 Sección de prueba

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

1

1.1. SECCIÓN DE PRUEBA CHAPTER 1. CAPÍTULO DE PRUEBA

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula.

2

Formato visual

Cuando tenemos terminado algún proyecto (puede ser un libro o un artículo), como paso final es gestionar las páginas en blanco o saltos de página. Para gestionar páginas, existen los siguientes comandos:

```
\pagebreak
\nopagebreak
\newpage
\clearpage
\cleardoublepage
```

\cleardoublepage

El comando `\cleardoublepage` hace que la página siguiente sea impar. Por ejemplo,

Ejemplo 5.15

```
\documentclass{book}
\usepackage{lipsum}
\begin{document}
  \chapter{Capítulo de prueba}
  \section{Sección de prueba}
  \lipsum[2]
  \cleardoublepage
  \lipsum[1]
\end{document}
```

Chapter 1

Capítulo de prueba

1.1 Sección de prueba

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

1

2 CHAPTER 1. CAPÍTULO DE PRUEBA

1.1. SECCIÓN DE PRUEBA 3

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Paquete titleps

El paquete titleps proporciona herramientas para el ajuste de estilos de página en una sola etapa (títulos y pie de página). Se utiliza una interfaz de nivel superior, donde se oculta el mecanismo de marcas y no es necesario tratar con `\leftmark` y `\rightmarks`, solo se utiliza un comando o variable registrada como "marca" como el valor esperado debido a que se devolverá el valor esperado, es decir, esos cuando se emitió la marca, ya sea mediante un comando de seccionamiento o explícitamente conm etc. Un ejemplo sencillo:

```
\newpagestyle{myestilo}{%
  \sethead[\thepage][\chaptertitle][\thesection]% even o par
          {\thesection}{\sectiontitle}{\thepage}% odd o impar
  \setfoot[\thepage][\chaptertitle][\thesection]% even o par
          {\thesection}{\sectiontitle}{\thepage}% odd o impar
}
```

Otras características:

1. Marcas superiores funcionando, compatibles con flotadores (a diferencia de la marca estándar `\topmark`, que no funciona correctamente en LaTeX).
2. Acceso a las marcas superiores, primero y último en un solo título/pie de página (por ejemplo, los números de sección primero y último).
3. Marcas para más de 2 niveles de seccionamiento.
4. Reglas de títulos y pies de página simples (y no tan simples).
5. Títulos y pies de página para páginas con flotadores.
6. Títulos y pies de página para flotadores específicos (una especie de `\thispagestyle` para flotadores).
7. Múltiples conjuntos de marcas (denominadas aquí conjuntos de marcas y marcas adicionales).

Este funcionalidades se pueden cargar por medio de una de las siguientes opciones:

```
\usepackage{titleps}
\usepackage[pagestyles]{titlesec}
```

Ejemplo 5.16

```
\documentclass{book}
\usepackage[x11names,table,svgnames,usenames,dvipsnames]{xcolor}
\usepackage[paperwidth=15cm, paperheight=23cm,margin=1.3cm]{geometry}
\usepackage[pagestyles,explicit]{titlesec}
\usepackage{lipsum}
\usepackage{tikz}
\usepackage{tikzpagenodes}
\definecolor{backgroundpage}{HTML}{C8B2D7}%dp1
\definecolor{fontpage}{HTML}{5F605D}
\newcommand\pagenumstyle{%
  \tikz{%
    \node[rectangle, rounded corners=3pt, fill=backgroundpage]{\bfseries\thepage};
  }
}
\newcommand\sectiontitlestyle{%
  \tikz{%
    \node[rectangle, rounded corners=3pt, fill=white]{\bfseries\thesection\;\;\sectiontitle};
  }
}
\newcommand\chaptertitlestyle{%
  \tikz{%
    \node[rectangle, rounded corners=3pt, fill=backgroundpage]{\bfseries\thechapter\;\;\chaptertitle};
  }
}
```

```
\newcommand\autortitlestyle{%
  \tikz{%
    \node[rectangle, rounded corners=3pt, fill=white]{\bfseries Heber MQ};
  }
}
\newpagestyle{myestilo}{%
  \sethead[\sectiontitlestyle][][\autortitlestyle]% even
          {\autortitlestyle}{}{\sectiontitlestyle}% odd impar
  \renewcommand\makeheadrule{%
    \begin{tikzpicture}[remember picture,overlay]
      \fill[backgroundpage] (current page.north west) rectangle ([xshift=2pt,yshift=-4mm]current
      ↪  page.east|-current page header area.south east);
    \end{tikzpicture}
  }
  \setfoot[\pagenumstyle][][\chaptertitlestyle]% even
          {\chaptertitlestyle}{}{\pagenumstyle}% odd impar
}
\newpagestyle{chapterpagestyle}{%
  \sethead[][][]% even
          {}{}{}% odd impar
  \renewcommand\makeheadrule{%
    \begin{tikzpicture}[remember picture,overlay]
      \fill[backgroundpage] (current page.north west) rectangle ([xshift=2pt,yshift=-4mm]current
      ↪  page.east|-current page header area.south east);
    \end{tikzpicture}
  }
  \setfoot[\pagenumstyle][][\chaptertitlestyle]% even
          {\chaptertitlestyle}{}{\pagenumstyle}% odd impar
}
\assignpagestyle{\chapter}{chapterpagestyle}
\begin{document}
  \pagestyle{myestilo}
  \chapter{Capítulo de prueba}
  \section{Sección de prueba}
  \lipsum[1-5]
\end{document}
```

Chapter 1

Capítulo de prueba

1.1 Sección de prueba

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis.

1 Capítulo de prueba 1

1.1 Sección de prueba Heber MQ

Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula.

Fusce mauris. Vestibulum luctus nibh at lectus. Sed bibendum, nulla a faucibus semper, leo velit ultricies tellus, ac venenatis arcu wisi vel nisl. Vestibulum diam. Aliquam pellentesque, augue quis sagittis posuere, turpis lacus congue quam, in hendrerit risus eros eget felis. Maecenas eget erat in sapien mattis porttitor. Vestibulum porttitor. Nulla facilisi. Sed a turpis eu lacus commodo facilisis. Morbi fringilla, wisi in dignissim interdum, justo lectus sagittis dui, et vehicula libero dui cursus dui. Mauris tempor ligula sed lacus. Duis cursus enim ut augue. Cras ac magna. Cras nulla. Nulla egestas. Curabitur a leo. Quisque egestas wisi eget nunc. Nam feugiat lacus vel est. Curabitur consectetuer.



2 1 Capítulo de prueba

TABLAS

LATEX tiene definidos dos entornos de tabulación o tablas: `tabular` y `array`. Además de estos entornos, existen otros paquetes como, por ejemplo, el tabularx y tabulary para realizar tablas. Estos paquetes tienen la opción de cálculos automáticos del ancho de la columna.

tabular y array

No se deben confundir los entornos tabulares y table. El entorno tabular organiza el texto de la tabla en columnas y el entorno table es un elemento de documento LATEX que tiene la capacidad de flotar. En el entorno table se pueden contener dos o más entornos tabulares.

tabular y table

El entorno estándar "table" de LATEX no permite hacer saltos de página o tener tablas de 2 o más páginas. Para crear tablas de 2 o más páginas es necesario recurrir a entornos como `longtable` o `supertabular`.

Tablas de dos o más páginas

Para realizar tablas a color, podemos utilizar el paquete tcolorbox.

6.1 Entornos estándares de LATEX

Los entornos estándares de LATEX para realizar tablas son: `tabbing`, `tabular` y `array`. Diferencias entre estos entornos:

1. El entorno `tabular` se puede personalizar de manera más sencilla que el entorno `tabbing`.
2. El entorno `tabbing` permite realizar saltos de página, algo que no sucede con el entorno `tabular`.
3. En el entorno `tabbing` se deben especificar las posiciones de tabulaciones y en el entorno tabular los anchos de las columnas se calculan de manera automática.

Entorno tabbing

En este entorno los comandos LATEX como `` \` `` y `\=` se utilizan para definir las posiciones de tabulaciones.

Comandos tabbing

El comando `\-` generalmente suele indicar una desconexión.

El comando `` \` `` (utiliza el parámetro `\tabbingsep`) es utilizado para alinear un texto a la derecha. El valor de predeterminado de `\tabbingsep` suele ser de 5 puntos.

El comando `\=` indica la posición de un tabulador con respecto al anterior.

El comando `\kill` es utilizado para finalizar una línea. Cuando se invoca este comando, las pestañas se restablecen.

El comando `\\` termina una línea tabulada y escribirla.

El comando `\>` avanza al siguiente tabulador o pestaña.

tabbing

```
\begin{tabbing}
  row1col1 \= row1col2 ...  \\
  row2col1 \> row2col2 ...  \\
  ...
\end{tabbing}
```

Ejemplo 6.1

```
\begin{tabbing}
  \hspace{1in}\=\hspace{0.8in}\=\kill
  Ship                 \>Guns                \>Year     \\
  \textit{Sophie}      \>14                  \>1800     \\
  \textit{Polychrest}  \>24                  \>1803     \\
  \textit{Lively}      \>38                  \>1804     \\
  \textit{Surprise}    \>28                  \>1805     \\
\end{tabbing}
```

Ship	Guns	Year
Sophie	14	1800
Polychrest	24	1803
Lively	38	1804
Surprise	28	1805

tabular

Entorno tabular

En este entorno las columnas se separan por `&` y las filas por `\\`.

Ejemplo 6.2

```
\documentclass{article}
\usepackage[paperwidth=10cm,paperheight=3cm, margin=1mm]{geometry}
\begin{document}
  \begin{table}
    \centering
    \begin{tabular}{cc}
      Nombre  & Puntaje   \\
      Alan    & 100       \\
      Bob     & 60        \\
      Chris   & 10        \\
      David   & 80
    \end{tabular}
    \caption{}
  \end{table}
\end{document}
```

Nombre	Puntaje
Alan	100
Bob	60
Chris	10
David	80

Table 1:

El entorno tabular tiene la siguiente sintaxis:

```
\begin{tabular}[<pos>]{<cols>}
  <text> & <text> & ... \\
  <text> & <text> & ...
\end{tabular}
\begin{tabular*}{<width>}[<pos>]{<cols>}
  <text> & <text> & ... \\
  <text> & <text> & ...
\end{tabular*}
```

El argumento `<cols>` es una secuencia de caracteres que indica el tipo de cada columna (ver tabla 6.12).

Código	
`l`	Alinea a la derecha
`c`	Centrea a la columna
`r`	Alinea a la izquierda
`p{<width>}`	Columna de ancho `<width>`
`|`	Añade una línea vertical
`@{decl}`	Elimina el espacio entre columnas.

Tabla 6.12

Tipos de columnas

Ejemplo 6.3

```
\begin{tabular}{|cc|c|}
  \hline
  p   &   q   &   p $\wedge$ q\\
  \hline
  v   &   v   &   v\\
  v   &   f   &   f\\
  f   &   v   &   f\\
  f   &   f   &   f\\
  \hline
\end{tabular}
```

p	q	$p \wedge q$
v	v	v
v	f	f
f	v	f
f	f	f

6.2 Referencia cruzada

`\ref`

`\label`

Para realizar referencias cruzadas se utiliza el comando `\ref` y `\label`.

Ejemplo 6.4

```
\documentclass{article}
\usepackage{geometry}
\geometry{paperwidth=10cm,paperheight=4cm,margin=1mm}
\usepackage{float}
\begin{document}
  \begin{table}[H]
    \centering
    \begin{tabular}{|cc|c|}
      \hline
      p   &   q   &   p $\wedge$ q\\
      \hline
      v   &   v   &   v\\
      v   &   f   &   f\\
      f   &   v   &   f\\
      f   &   f   &   f\\
      \hline
    \end{tabular}
    \caption{Tabla de prueba}
    \label{tab:tablaVerdad}
  \end{table}
  En la tabla \ref{tab:tablaVerdad} se muestra la tabla de verdad.
\end{document}
```

p	q	p $\wedge$ q
v	v	v
v	f	f
f	v	f
f	f	f

Table 1: Tabla de prueba

En la tabla 1 se muestra la tabla de verdad.

|, \hline y \cline

6.3 Líneas horizontales y verticales

Para añadir líneas verticales se debe utilizar las marcas `|`. Para añadir una línea horizontal se debe invocar el comando `\hline`. Para añadir una línea desde la columna `i` hasta `j` se debe recurrir al comando `\cline{i-j}`.

Ejemplo 6.5

```
\begin{tabular}{|c|c|c|}
  \hline
  $p$ & $q$ & $p \rightarrow q$ \\ \hline
  0 & 0 & 1 \\
  0 & 1 & 1 \\ \cline{1-2}
  1 & 0 & 0 \\
  1 & 1 & 1 \\ \hline
\end{tabular}
```

p	q	$p \rightarrow q$
0	0	1
0	1	1
1	0	0
1	1	1

También existe el entorno tabular con estrella, solo que en este caso el ancho ocupado por la tabla se debe especificar.

tabular*

```
\begin{tabular*}{<width>}[<pos>]{<cols>}
  <text> & <text> & ... \\
  <text> & <text> & ...
\end{tabular*}
```

Ejemplo 6.6

```
\begin{tabular*}{5cm}{|cc|c|}
  \hline
  p   &   q   &   p $\wedge$ q\\
  \hline
  V   &   V   &   V\\
  V   &   F   &   F\\
  F   &   V   &   F\\
  F   &   F   &   F\\
  \hline
\end{tabular*}
```

p	q	p $\wedge$ q	
V	V	V	
V	F	F	
F	V	F	
F	F	F	

También existe el entorno array para realizar tablas en LaTeX, sólo que en este caso el contenido debe ser expresiones matemáticas (se estudiará en el capítulo de modo matemático).

Parámetros para personalizar la apariencia de una tabla:

1. `\arraycolsep`: Establece el espacio entre columnas en el entorno `array`.
2. `\tabcolsep`: Establece el espacio entre columnas en el entorno `tabular`.
3. `\arrayrulewidth`: Establece el ancho de las líneas verticales.
4. `\doublerulesep`: Establece el espacio entre dos caracteres, `||`, creados.
5. `\arraystretch`: Establece espacio vertical de las líneas.

Ejemplo 6.7

```
\documentclass{article}
\usepackage[paperwidth=12cm,paperheight=5cm,margin=1mm]{geometry}
\setlength{\tabcolsep}{1cm}
\setlength{\arrayrulewidth}{2pt}
\renewcommand*{\arraystretch}{2}
\begin{document}
  \centering
  \begin{tabular}{|cc|c|}
    \hline
    p   &   q   &   p $\wedge$ q\\
    \hline
    V   &   V   &   V\\
    V   &   F   &   F\\
    F   &   V   &   F\\
    F   &   F   &   F\\
    \hline
  \end{tabular}
\end{document}
```

p	q	$p \wedge q$
V	V	V
V	F	F
F	V	F
F	F	F

6.4 Paquete array

array

El paquete `array` adiciona algunas funcionalidades.

Código	
`|`	Inserta una línea vertical
`m{width}`	Define el ancho de una columna.
`>{decl}`	Inserta `<decl>` antes de entrar a la columna.
`<{decl}`	Inserta `<decl>` después de entrar a la columna.
`!{decl}`	Es similar que `|`; solo que se ejecuta `decl` en vez de `|`.

Tabla 6.13

Opciones adicionales del paquete array

Para obtener toda una columna en negrillas podemos usar `>{\bfseries}c`.

Ejemplo 6.8

```
\begin{tabular}{|>{\bfseries}cc|c|}
  \hline
  p   &   q   &   p $\wedge$ q\\
  \hline
  V   &   V   &   V\\
  V   &   F   &   F\\
  F   &   V   &   F\\
  F   &   F   &   F\\
  \hline
\end{tabular}
```

p	q	$p \wedge q$
V	V	V
V	F	F
F	V	F
F	F	F

Diferencias entre las opciones `p{width}`, `m{width}` y `b{width}`:

Ejemplo 6.9

```
\begin{tabular}{|p{1cm}|p{1cm}|p{1cm}|}
  \hline
  a a a a a  a a a a a a a a a a a & b b b b b b b b b b b
  ↪  & c c c c \\
  \hline
\end{tabular}
```

a a a a a a a a a a a a a a a a	b b b b b b b b b b b	c c c c

Ejemplo 6.10

```
\begin{tabular}{|m{1cm}|m{1cm}|m{1cm}|}
  \hline
  a a a a a  a a a a a a a a a a a & b b b b b b b b b b b
  ↪  & c c c c \\
  \hline
\end{tabular}
```

a a a a a a a a a a a a a a a a	b b b b b b b b b b b	c c c c

Ejemplo 6.11

```
\begin{tabular}{|b{1cm}|b{1cm}|b{1cm}|}
  \hline
  a a a a a  a a a a a a a a a a a & b b b b b b b b b b b
  ↪  & c c c c \\
  \hline
\end{tabular}
```

a a a a a a a a a a a a a a a a	b b b b b b b b b b b	c c c c

Todas las opciones `p{width}`, `m{width}` y `b{width}` tienen la opción `\parident` igual a `0pt`, pero esto se puede cambiar de la siguiente forma:

Ejemplo 6.12

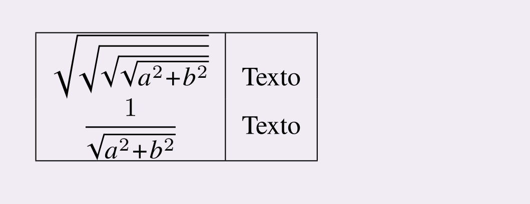
```
\begin{tabular}{|>{\setlength\parindent{5mm}}p{2cm}|p{2cm}|}
  \hline
  a a a a a a a a a a a a a a a a a a a a a & b b b b b b b b b b
  ↪ b \\
  \hline
\end{tabular}
```

a a a a a a a a a a a a a a a a a a a a a	b b b b b b b b b b b

Para que una columna se inicialice y finalice el modo matemático, de una manera automática, podemos recurrir al código `>{$}c<{$}`.

Ejemplo 6.13

```
\begin{tabular}{|>{$}c<{$}|c|}
  \hline
  \sqrt{\sqrt{\sqrt{\sqrt{a^2+b^2}}}} & Texto\\
  \dfrac{1}{\sqrt{a^2+b^2}} & Texto\\
  \hline
\end{tabular}
```



Para dividir o romper las palabras se puede recurrir al comando `\hspace{0pt}` que el cual se puede combinar con el comando `\raggedright`, pero en vez del comando `\\` se debe utilizar `\tabularline`.

Ejemplo 6.14

```
\begin{tabular}{|>{\hspace{0pt}}p{1cm}|c|}
  \hline
  TextoTextoTexto & Texto\\
  \hline
\end{tabular}
```

Texto-Texto-Texto	Texto

Ejemplo 6.15

```
\begin{tabular}{|>{\hspace{0pt}\raggedright}p{1.5cm}|c|}
  \hline
  TextoTextoTexto TextoTextoTexto & Texto\\
  \hline
\end{tabular}
```

TextoTex-toTexto TextoTex-toTexto	Texto

Espacio entre columnas

El espacio entre columnas está definido por `\arraycolsep` para el entorno `array` y `\tabcolsep` para el entorno `tabular`. Al modificar estos parámetros, esto afecta a todas las columnas de una tabla, pero para modificar el espacio entre dos columnas podemos recurrir al comando `@{}`.

Ejemplo 6.16

```
\begin{tabular}{c@{}c!{}c@{--}c@{\hspace{1cm}}c}
  \hline
  Celda 1 & Celda 2 & Celda 3 & Celda 4 & Celda 5\\
  1 & 2 & 3 & 4 & 5\\
  \hline
\end{tabular}
```

Celda 1	Celda 2	Celda 3–	Celda 4	Celda 5
1	2	3 –	4	5

1. El código `@{}` elimina el espacio entre columnas.
2. El código `!{espacio}` no tiene ningún efecto.
3. El código `@{espacio}` añade la cadena “espacio” entre las columnas 3 y 4.
4. El código `@{\hspace{2.5cm}}` añade un espacio de 2.5 cm entre las columnas 4 y 5.

Nuevas columnas

Hasta el momento hemos utilizado los tipos columnas como `c`, `l`, `r` y entre otros. Para definir un nuevo tipo de columna debemos de utilizar el comando `\newcolumntype`.

`\newcolumntype`

```
\newcolumntype{<name column>}[<arguments>]{<decl>}
```

En donde `<name column>` es el nombre de la columna (no debe ver nombres duplicados y además se recomienda que sea solo una letra), `<arguments>` es el número de argumentos y `<decl>` es la declaración de la columna.

Definir nuevo tipo de columna

Podemos definir nuevos tipos de columnas para el modo matemático como se muestra en el ejemplo.

Ejemplo 6.17

```
\newcolumntype{C}{>{$}c<{$}}
\newcolumntype{L}{>{$}l<{$}}
\newcolumntype{R}{>{$}r<{$}}
\begin{tabular}{|L|R|C|}
  \hline
  p   &   q   &   p \wedge q\\
  \hline
  V   &   V   &   V\\
  V   &   F   &   F\\
  F   &   V   &   F\\
  F   &   F   &   F\\
  \hline
\end{tabular}
```

p	q	$p \wedge q$
V	V	V
V	F	F
F	V	F
F	F	F

6.5 Paquete tabularx y tabulary

El paquete `tabularx` está implementado sobre el entorno `tabular`, en el cual se calculan los anchos de las columnas de manera automática. El entorno `tabularx` proporcionado por el paquete `tabularx` hereda los parámetros del entorno `tabular*`.

tabularx

Sintaxis del entorno `tabularx`:

```
\begin{tabularx}{<width>}{<column specification>}
  <table content>
\end{tabularx}
```

Ejemplo 6.18

```
\begin{tabularx}{5cm}{ccc}
  \hline
  Celda 1 & Celda 2 & Celda 3\\
  \hline
\end{tabularx}
```

Celda 1	Celda 2	Celda 3

Diferencias entre tabular*, tabularx y tabulary

1. `tabularx` y `tabulary` cambian el ancho de la columna, mientras que el entorno `tabular*` sólo cambia el espacio entre las columnas.

2. Dentro de los entornos `tabularx` y `tabulary`, el comando `\verb` se parcializa debido a que no se puede manejar correctamente los espacios.

3. El entorno `tabular*` gestiona los espacios utilizando las primitivas TEX, mientras que los entornos `tabularx` y `tabulary` formatean recursivamente hasta obtener una columna óptima; por esta razón, estos entornos podrían tardar al momento de compilar el documento LATEX.

6.6 Tablas de varias páginas

Ahora nos enfocaremos en crear tablas que abarquen más de una página. Los entornos estudiados anteriormente no tienen la posibilidad de romperse (o dividirse) en partes de tal manera que abarquen dos o más páginas. Los paquetes supertabular y longtable proporcionan entornos para realizar tablas de dos o más páginas.

Supertabular

supertabular

Sintaxis de los entornos proporcionados por el paquete supertabular:

```
\begin{supertabular}{<cols>}
  rows
\end{supertabular}
\begin{supertabular*}{<width>}{<cols>}
  rows
\end{supertabular*}
\begin{mpsupertabular}{<cols>}
  rows
\end{mpsupertabular}
\begin{mpsupertabular*}{width}{<cols>}
  rows
\end{mpsupertabular*}
```

En la documentación de este paquete, el entorno `supertabular` se define utilizando el entorno `tabular` y el entorno `supertabular*` bajo el entorno `tabular*`. Cuando se detecta que la altura de la tabla es igual a `\textheight`, entonces se agrega `\end{tabular}` (también inicializa el `\begin{tabular}`) y de esta forma la tabla se divide.

Como el paquete `supertabular` utiliza el entorno `tabular`, entonces hereda la mayoría de sus funciones del entorno tabular `tabular`.

Comandos proporcionados por el paquete `supertabular`:

Parámetros de supertabular

1. `\tablefirsthead`: Este comando define el contenido del encabezado de la tabla.
2. `\tablehead`: Define el contenido en el encabezado de la tabla y se repite en cada página.
3. `\tabletail`: Es utilizado para insertar algún código antes del `\end{tabular}`.
4. `\tablelasttail`: Es utilizado para insertar algún código después del último `\end{tabular}`.
5. `\topcaption`: Los comandos `\topcaption`, `\bottomcaption` y `\tablecaption` se comportan de igual manera que el comando `\caption` y además tienen el mismo objetivo, es decir, son utilizados para definir el título de la tabla.
6. `\bottomcaption`.
7. `\tablecaption`.

Ejemplo 6.19

```
\documentclass{article}
\usepackage{geometry}
\geometry{paperwidth=18cm,paperheight=20cm,margin=1cm}
\usepackage{supertabular,multirow,booktabs}
\usepackage{lipsum}
\begin{document}
```

```latex
  \lipsum[1-3]

  \begin{supertabular}{p{2.8cm}p{0.6cm}p{1.6cm}p{2.2cm}p{0.8cm}p{2.6cm}p{2.5cm}}
    \toprule
    \textbf{Name} & \textbf{Year} & \textbf{ID} & \textbf{Address} & \textbf{Salary} & \textbf{Skills}
    ↪  & \textbf{Qualifications} \\
    \midrule
    Some text & some text & some text & 5 & some text & Som text &\\
    Some text & some text & some text & 5 & some text & Som text &\\
    Some text & some text & some text & 5 & some text & Som text &\\
    Some text & some text & some text & 5 & some text & Som text &\\
    Some text & some text & some text & 5 & some text & Som text &\\
    Some text & some text & some text & 5 & some text & Som text &\\
    Some text & some text & some text & 5 & some text & Som text &\\
    Some text & some text & some text & 5 & some text & Som text &\\
    Some text & some text & some text & 5 & some text & Som text &\\
    Some text & some text & some text & 5 & some text & Som text &\\
    Some text & some text & some text & 5 & some text & Som text &\\
    Some text & some text & some text & 5 & some text & Som text &\\
    \bottomrule
  \end{supertabular}
\end{document}
```

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

Name	**Year**	**ID**	**Address**	**Salary**	**Skills**	**Qualifications**
Some text	some text	some text	5	some text	Som text	
Some text	some text	some text	5	some text	Som text	
Some text	some text	some text	5	some text	Som text	
Some text	some text	some text	5	some text	Som text	
Some text	some text	some text	5	some text	Som text	
Some text	some text	some text	5	some text	Som text	
Some text	some text	some text	5	some text	Som text	
Some text	some text	some text	5	some text	Som text	
Some text	some text	some text	5	some text	Som text	

1

Some text	some text	some text	5	some text	Som text	
Some text	some text	some text	5	some text	Som text	
Some text	some text	some text	5	some text	Som text	

2

Longtable

Este paquete, a diferencia del paquete `supertabular`, el entorno `longtable` (este entorno es proporcionado por el paquete `longtable`) almacena el ancho de cada entorno en un archivo auxiliar con extensión `.aux`.

Sintaxis del entorno `longtable`:

longtable

```
\begin{longtable}[<align>]{<cols>}
  rows
\end{longtable}
```

Este sintaxis es similar al sintaxis del entorno tabular. Valores que pueda adoptar el argumento obligatorio: `c`, `l` y `r`.

La mayoría de los comandos proporcionados por el paquete `longtable` se invocan dentro del entorno `longtable`.

Comandos proporcionados por el paquete `longtable`:

1. `\caption[]{}`. Este comando es equivalente al comando `\caption` del entorno `table`.
2. `\label{}`. Este comando es equivalente al comando `\label` del entorno `table`.
3. `\endfirsthead`. Crea el encabezado principal de la tabla.
4. `\endhead`. Define el encabezado de la tabla ubicado en la siguiente página.
5. `\endfoot`. Define el pie de la tabla, y se muestra en todas las páginas (excepto la en la última).
6. `\endlastfoot`. Define el pie de la tabla, y se muestra en la última página.

Cuando se utiliza el paquete `caption` el comando `\caption` entra en conflicto, por tal razón debemos descubrir al comando `\captionof{}{}`.

`\captionof`

Ejemplo 6.20

```
\documentclass{article}
\usepackage{geometry}
\geometry{paperwidth=18cm,paperheight=19cm,margin=1cm}
\usepackage{longtable,multirow,booktabs}
\usepackage{lipsum}
\begin{document}
  \lipsum[1-4]
  \begin{longtable}{lcccccr}
    \caption{Tabla de más de una página}\\
    \toprule
    \textbf{Name} & \textbf{Year} & \textbf{ID} & \textbf{Address} & \textbf{Salary} & \textbf{Skills}
     ↪ & \textbf{Qualifications} \\
```

```
    \endfirsthead
    \midrule
    Some text & some text & some text & 5 & some text & Som text &\\
    Some text & some text & some text & 5 & some text & Som text &\\
    Some text & some text & some text & 5 & some text & Som text &\\
    Some text & some text & some text & 5 & some text & Som text &\\
    Some text & some text & some text & 5 & some text & Som text &\\
    Some text & some text & some text & 5 & some text & Som text &\\
    Some text & some text & some text & 5 & some text & Som text &\\
    Some text & some text & some text & 5 & some text & Som text &\\
    Some text & some text & some text & 5 & some text & Som text &\\
    Some text & some text & some text & 5 & some text & Som text &\\
    Some text & some text & some text & 5 & some text & Som text &\\
    Some text & some text & some text & 5 & some text & Som text &\\
    Some text & some text & some text & 5 & some text & Som text &\\
    \bottomrule
  \end{longtable}
\end{document}
```

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula.

Table 1: Tabla de más de una página

Name	**Year**	**ID**	**Address**	**Salary**	**Skills**	**Qualifications**
Some text	some text	some text	5	some text	Som text	
Some text	some text	some text	5	some text	Som text	
Some text	some text	some text	5	some text	Som text	
Some text	some text	some text	5	some text	Som text	
Some text	some text	some text	5	some text	Som text	
Some text	some text	some text	5	some text	Som text	
Some text	some text	some text	5	some text	Som text	
Some text	some text	some text	5	some text	Som text	

1

Some text	some text	some text	5	some text	Som text	
Some text	some text	some text	5	some text	Som text	
Some text	some text	some text	5	some text	Som text	
Some text	some text	some text	5	some text	Som text	
Some text	some text	some text	5	some text	Som text	

2

6.7 Tablas a colores

Para colorear las tablas debemos de utilizar el paquete color.

Ejemplo 6.21

```
\begin{tabular}{>{\color{blue!50!white}\bfseries}lr}
  Estudiante &\textcolor{red!50!white}{\bfseries Nota}\\ \hline
  Juan & 10\\
  Pedro & 9\\
  Ana & 8
\end{tabular}
```

Estudiante	Nota
Juan	10
Pedro	9
Ana	8

Para colorear el fondo de una columna debemos de recurrir al paquete `colortbl`:

Ejemplo 6.22

```
\begin{tabular}{>{\columncolor{blue!50!white}\color{white}\bfseries}lr}
  \rowcolor[gray]{0.8}
  \color{black} Estudiante & \bfseries Nota\\ \hline
  Juan & 10\\
  Pedro & 9\\
  Ana & 8\\
  \cellcolor[gray]{0.8}\color{black}
  Promedio & 9
\end{tabular}
```

Estudiante	Nota
Juan	10
Pedro	9
Ana	8
Promedio	9

6.8 Personalizando espacios y líneas

Líneas a colores

Los colores de las líneas podemos gestionar con el paquete de `colortbl`, este paquete proporciona ciertos comandos para cambiar los colores de las líneas de una tabla.

Los colores de las líneas pueden cambiarse globalmente o sólo para un entorno específico. Para realizar los cambios globalmente debemos de realizar las configuraciones respectivas en el preámbulo del documento LaTeX y para cambiar para un entorno específico debemos de realizar las configuraciones antes de invocar el entorno.

Ejemplo 6.23

```
\documentclass{article}
\usepackage[paperwidth=10cm,paperheight=3cm,margin=1mm]{geometry}
\usepackage{xcolor,colortbl}
\setlength{\arrayrulewidth}{0.5pt}
\newcommand{\myhline}{\arrayrulecolor{red!50!white}\hline\arrayrulecolor{black}}
\newcommand{\mycline}[1]{\arrayrulecolor{blue!50!white}\cline{#1}\arrayrulecolor{black}}
\begin{document}
  \begin{table}
    \centering
    \begin{tabular}{|c|c|c|}
```

```
      \myhline
      $p$ & $q$ & $p \rightarrow q$ \\ \myhline
      0 & 0 & 1 \\
      0 & 1 & 1 \\ \mycline{1-2}
      1 & 0 & 0 \\
      1 & 1 & 1 \\ \myhline
    \end{tabular}
    \caption{}
  \end{table}
\end{document}
```

p	q	$p \rightarrow q$
0	0	1
0	1	1
1	0	0
1	1	1

Table 1:

Espacios entre las filas

Anteriormente ya habíamos utilizado los comandos `\arraystretch` y `\extrarowheight` para modificar el espacio entre filas de una tabla, pero al momento de modificar estos comandos todos los espacios entre las filas se modifican; por tanto, para modificar específicamente el espacio entre 2 líneas debemos de recurrir a ciertos paquetes como `tabls`.

Macros de tabls

Comandos proporcionados por el paquete `tabls`:

1. `\tablinesep`. Define el espacio mínimo entre líneas.
2. `\arraylinesep`. Es equivalente a `\tablinesep`.
3. `\extrarulesep`. Adiciona un espacio encima y debajo de cada línea definido por `\hline`.

Ejemplo 6.24

```
\documentclass{article}
\usepackage{geometry,tabls,float}
\geometry{paperwidth=10cm,paperheight=3cm,margin=1mm}
\begin{document}
  \setlength\tablinesep{10pt}
  \begin{table}[H]
    \centering
    \begin{tabular}{|c|c|c|}
      \hline
      $p$ & $q$ & $p \rightarrow q$ \\ \hline
      0 & 0 & 1 \\
```

```
    0 & 1 & 1 \\
    1 & 0 & 0 \\
    1 & 1 & 1 \\ \hline
  \end{tabular}
  \caption{}
 \end{table}
\end{document}
```

p	q	$p \rightarrow q$
0	0	1
0	1	1
1	0	0

Paquete booktabs

Este paquete tiene otro enfoque al momento de realizar tablas, las tablas con este paquete solo tienen líneas horizontales; por tanto, este paquete proporciona los comandos:

Macros booktabs

1. `\toprule[<width>]`. Crea una línea en la parte superior de la tabla.

2. `\midrule[<width>]`. Crea una línea entre dos fílas de la tabla.

3. `\bottomrule[<width>]`. Crea una línea en la parte inferior de la tabla.

Ejemplo 6.25

```
\begin{tabular}{ccc}
  \toprule
  $p$ & $q$ & $p \rightarrow q$ \\ \midrule
  0 & 0 & 1 \\
  0 & 1 & 1 \\
  1 & 0 & 0 \\
  1 & 1 & 1 \\ \bottomrule
\end{tabular}
```

p	q	$p \rightarrow q$
0	0	1
0	1	1
1	0	0
1	1	1

4. `\cmidrule[<width>](<trim>){<col1-col2>}`. Este comando tiene un comportamiento similar al comando `\cline`.

Ejemplo 6.26

```
\begin{tabular}{ccc}
  \toprule
  $p$ & $q$ & $p \rightarrow q$ \\ \midrule
  0 & 0 & 1 \\
  0 & 1 & 1 \\ \cmidrule{1-2}
  1 & 0 & 0 \\
  1 & 1 & 1 \\ \bottomrule
\end{tabular}
```

p	q	$p \rightarrow q$
0	0	1
0	1	1
1	0	0
1	1	1

6.9 Alineación vertical

Para poner un texto en una columna, entera o parcial, de una tabla de varias filas se debe recurrir al paquete `multirow`. Este paquete proporciona el comando

```
\multirow{<nrow>}[<njot>]{<width>}[<vmove>]{<contents>}
```

Ejemplo 6.27

```
\begin{tabular}{|c|c|c|c|}
  \hline
  \multirow{3}{2cm}{test text, test text, test text}
    &test text & \multirow{2}{1.5cm}{test text} & test text \\
    &test text &                                 & test text \\
    &test text & test text                       & test text \\ \hline
\end{tabular}
```

test text, test text, test text	test text	test text	test text
	test text		test text
	test text	test text	test text

6.10 Unir celdas de una tabla

Para unir dos o más celdas en una tabla, debemos de recurrir al comando

```
\multicolumn{<ncolums>}{<align>}{<contents>}
```

En donde `<ncolums>` es el número de columnas que se quiere unir, `<align>` es el alineamiento de la celda (puede ser `l`, `c` y `r`), y `<contents>` es el contenido de la celda.

Ejemplo 6.28

```
\begin{center}
  \begin{tabular}{|lcc|lcc|lcc|lcc|}\hline
    \multicolumn{3}{|c|}{Monovalentes} & \multicolumn{3}{c|}{Bivalentes} &
    ↪  \multicolumn{3}{c|}{Trivalentes}      & \multicolumn{3}{c|}{Bi-trivalentes}   \\ \hline
    Litio    & Li        & $+1$           & Berilio    & Be & $+2$            & Aluminio & Al & $+3$
    ↪  & Hierro  & Fe & $+2,+3$                 \\
    Sodio    & Na        & $+1$           & Magnesio   & Mg & $+2$            & Galio    & Ga & $+3$
    ↪  & Niquel  & Ni & $+2,+3$                 \\
    Potasio & K          & $+1$           & Calcio     & Ca & $+2$            & Indio    & In & $+3$
    ↪  & Cobalto & Co & $+2,+3$                 \\ \cline{7-9} \cline{10-12}
    Rubidio & Rb         & $+1$           & Estroncio & Sr & $+2$            &
    ↪  \multicolumn{3}{c|}{Mono-bivalentes}  & \multicolumn{3}{c|}{Bi-tetravalentes} \\ \cline{7-9}
    ↪  \cline{10-12}
    Cesio    & Cs        & $+1$           & Bario      & Ba & $+2$            & Cobre    & Cu & $+1,+2$
    ↪  & Estaño  & Sn & $+2,+4$                 \\
    Francio & Fr         & $+1$           & Radio      & Ra & $+2$            & Mercurio & Hg & $+1,+2$
    ↪  & Plomo   & Pb & $+2,+4$                 \\ \cline{7-9}
    Plata    & Ag        & $+1$           & Zinc       & Zn & $+2$            &
    ↪  \multicolumn{3}{c|}{Mono-trivalentes} & Paladio & Pd & $+2,+4$                 \\ \cline{7-9}
    Amonio   & \ce{NH4} & $+1$           & Cadmio     & Cd & $+2$            & Oro      & Au & $+1,+3$
    ↪  & Platino & Pt & $+2,+4$                 \\
            &            &                &            &    &                 & Talio    & Tl & $+1,+3$
            ↪  &          &    &                        \\\hline
  \end{tabular}
\end{center}
```

Monovalentes			Bivalentes			Trivalentes			Bi-trivalentes		
Litio	Li	+1	Berilio	Be	+2	Aluminio	Al	+3	Hierro	Fe	+2,+3
Sodio	Na	+1	Magnesio	Mg	+2	Galio	Ga	+3	Niquel	Ni	+2,+3
Potasio	K	+1	Calcio	Ca	+2	Indio	In	+3	Cobalto	Co	+2,+3
Rubidio	Rb	+1	Estroncio	Sr	+2	Mono-bivalentes			Bi-tetravalentes		
Cesio	Cs	+1	Bario	Ba	+2	Cobre	Cu	+1,+2	Estaño	Sn	+2,+4
Francio	Fr	+1	Radio	Ra	+2	Mercurio	Hg	+1,+2	Plomo	Pb	+2,+4
Plata	Ag	+1	Zinc	Zn	+2	Mono-trivalentes			Paladio	Pd	+2,+4
Amonio	NH_4	+1	Cadmio	Cd	+2	Oro	Au	+1,+3	Platino	Pt	+2,+4
						Talio	Tl	+1,+3			

FLOTADORES

Cuando se compila un documento LaTeX, el sistema TeX divide de manera automática el texto en páginas de tal modo que ocupe la menor cantidad de páginas posibles, pero no se pueden dividir las figuras y las tablas, por lo que éstas se manejan como objetos flotantes. Los entornos como `figure` y `table` son entornos flotantes.

Objetos flotantes

Un objeto flotante se guarda en una caja y esta caja se guarda dentro de una lista para luego ser desempaquetado la lista bajo un algoritmo.

LaTeX ha sido construido para documentos que contienen poca cantidad de gráficas, por lo que en aquellos libros y artículos que contengan muchos gráficos, tendrán tendencia a desordenarse deslizándose o flotando hacia la parte final del documento.

Cuando se tienen varios objetos flotantes en un documento LaTeX y se invoca el comando `\clearpage` en un determinado lugar del documento LaTeX, entonces este comando hace que se finalice la página actual y elimine todos los objetos flotantes que se están desplazando, es decir, hace que se impriman de manera obligatoria los objetos flotantes que aparecen antes de invocar el comando. Los objetos flotantes se colocan en una página adicional. Cuando se tienen documentos de dos caras, este comando hace que la siguiente página de contenido sea impar, entonces si termina en una página impar, agrega una página en blanco para que comience con una página impar.

`\clearpage`

Ejemplo 7.1

```
\documentclass{article}
\usepackage{graphicx,geometry,lipsum,pdfpages}
\geometry{paperwidth=12cm, paperheight=13cm, margin=5mm}
\begin{document}
\section{Sección 1}
\lipsum[1]
\begin{figure}
  \centering
  \includegraphics[width=3cm]{images/grafica1}
  \caption{Figura 1}
\end{figure}
\begin{figure}
  \centering
```

```
    \includegraphics[width=3cm]{images/grafica1}
    \caption{Figura 2}
  \end{figure}
  \begin{figure}
    \centering
    \includegraphics[width=3cm]{images/grafica1}
    \caption{Figura 3}
  \end{figure}
  \lipsum[2]
  \end{document}
```

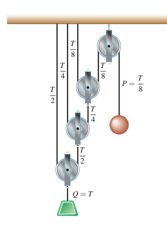


Figure 1: Figura 1

1 Sección 1

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum. Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis

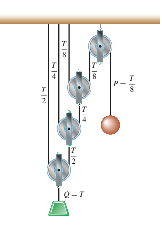


Figure 2: Figura 2

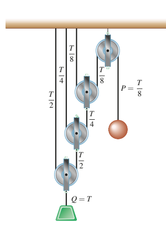


Figure 3: Figura 3

vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Ejemplo 7.2

```
\documentclass{article}
\usepackage{graphicx,geometry,lipsum,pdfpages}
\geometry{paperwidth=12cm, paperheight=13cm, margin=5mm}
\begin{document}
\section{Sección 1}
\lipsum[1]
\begin{figure}
  \centering
  \includegraphics[width=3cm]{images/grafica1}
  \caption{Figura 1}
\end{figure}
\begin{figure}
  \centering
  \includegraphics[width=3cm]{images/grafica1}
  \caption{Figura 2}
\end{figure}
\begin{figure}
  \centering
  \includegraphics[width=3cm]{images/grafica1}
  \caption{Figura 3}
\end{figure}
\clearpage
```

```
\lipsum[2]
\end{document}
```

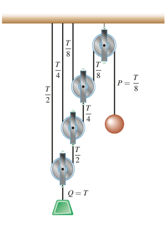


Figure 1: Figura 1

1 Sección 1

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

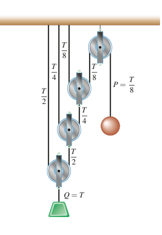


Figure 2: Figura 2

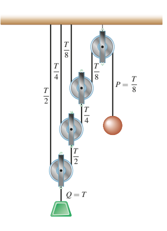


Figure 3: Figura 3

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

7.1 Propiedades de objetos flotantes

Para imprimir los objetos flotantes en el lugar en el que se invocan los entornos, debemos recurrir a ciertos paquetes como `float` y, de la misma forma, también podemos ajustar los parámetros del flotante.

Utilizaremos el término "flotante" para referirnos a un objeto flotante como una tabla o figura y "página flotante" para páginas que solamente contienen flotantes.

Algunos parámetros de objetos flotantes:

Parámetros de objetos flotantes

1. `topnumber`: Es un contador que especifica el número de flotantes que se van a imprimir en la parte superior de la página (su valor predeterminado es 2).
2. `bottomnumber`: Es un contador que especifica el número de flotantes que se van a imprimir en la parte inferior de la página (su valor predeterminado es 1).
3. `totalnumber`: Es un contador que especifica el número de flotantes que se van a imprimir en la página (su valor predeterminado es 3).

Cuando se tienen varios objetos flotantes, el algoritmo de LaTeX llena una página con los objetos flotantes y luego se detiene este algoritmo para continuar con la siguiente página, llenando con los flotantes restantes. La creación de páginas como objetos flotantes está controlada por el parámetro `\floatpagefraction`.

Si realizamos la siguiente operación:

```
\renewcommand{\floatpagefraction}{.8}
```

Esto hará que el algoritmo de LaTeX imprima el 80% de la página con flotantes y el 20% con texto.

7.2 Posicionamiento de objetos flotantes

El algoritmo de LaTeX posiciona los objetos flotantes en la parte superior de la página, pero algunas veces es necesario que nuestro objeto flotante aparezca en el lugar en el que invocamos nuestro entorno flotante. Por ello, podemos añadir `!` al argumento opcional del entorno flotante.

Ejemplo 7.3

```
\documentclass{article}
\usepackage{graphicx,geometry,lipsum,pdfpages}
\geometry{paperwidth=12cm, paperheight=17cm, margin=5mm}
\begin{document}
\section{Sección 1}
\lipsum[1]
\begin{figure}[h!]
  \centering
  \includegraphics[width=3cm]{images/grafica1}
  \caption{Figura 1}
\end{figure}
```

```
\lipsum[1]
\end{document}
```

1 Sección 1

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut

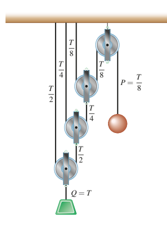


Figure 1: Figura 1

purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Esta operación no se cumple con las páginas flotantes.

El caracter `!` indica que las restricciones impuestas por los parámetros flotantes sea ignorado.

Entorno flotante

Sintaxis de los entornos flotantes `figure` y `table`:

```
\begin{nombreEntorno}[<posicionamiento>]
  ...
\end{nombreEntorno}
```

Posicionamiento de objeto flotante

El argumento opcional `<posicionamiento>` puede adoptar los valores `h`, que indica que se sitúe el objeto flotante en el mismo lugar donde se invoca el entorno (en inglés se interpreta como "here"); `t`, que indica que se ubique en la parte superior de la página (top); `b`, que indica que se sitúe en la parte inferior de la página (bottom); `p` que indica que se sitúe en una columna o página flotante; y `H`, que indica que se sitúe en el mismo lugar (HERE) y es proporcionado por el paquete `float`. Los argumentos mencionados (excepto `H`) se pueden combinar; por ejemplo, `bt` (es lo mismo que `tb`) indica que se sitúe el objeto flotante en la parte superior o inferior. En este caso, el algoritmo de LaTeX tiene preferencias por situar los objetos

flotantes en la parte superior de la página, por lo que primero va a tratar de ubicarlos en la parte superior de la página y luego en la parte inferior.

\FloatBarrier

Paquete placeins

Este paquete hace que los flotadores no pasen más allá del comando \FloatBarrier, generalmente este comando suele utilizarse para que no pasen los flotadores a la siguiente sección.

Ejemplo 7.4

```
\documentclass{article}
\usepackage{lipsum,geometry,pdfpages}
\geometry{paperwidth=12cm,paperheight=17cm,margin=5mm}
\begin{document}
\lipsum[1]
\section{Sección}
\begin{figure}[tb]
  \centering
  \includegraphics[width=3cm]{images/grafica1}
  \caption{Figura 1}
\end{figure}
\lipsum[1]
\end{document}
```

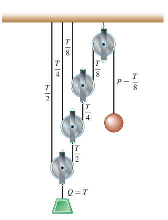


Figure 1: Figura 1

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

1 Sección

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Ejemplo 7.5

```
\documentclass{article}
\usepackage{placeins,lipsum,geometry,pdfpages}
\geometry{paperwidth=12cm,paperheight=17cm,margin=5mm}
\begin{document}
\lipsum[1]
\FloatBarrier
\section{Sección}
\begin{figure}[tb]
  \centering
  \includegraphics[width=3cm]{images/grafica1}
  \caption{Figura 1}
\end{figure}
\lipsum[1]
\FloatBarrier
\end{document}
```

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

1 Sección

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent

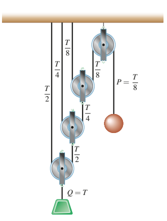


Figure 1: Figura 1

eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

También podemos indicar a este paquete que no pase los flotantes al siguiente sección utilizando `section` como argumento opcional al momento de llamar el paquete `placeins`, es decir,

```
\usepackage[section]{placeins}
```

Paquete afterpage

Este paquete proporciona el comando `\afterpage` y hace que los comandos especificados se expandan después de generar la página actual.

Cuando invocamos el comando `\clearpage` éste finaliza la página de manera no deseada, pero en vez de ello nosotros podemos invocar el código

```
\afterpage{\clearpage}
```

Este comando hará que se llene de texto la página actual y luego eliminará a todos los objetos flotantes cuando comience la siguiente página.

7.3 Paquete float

Los entornos flotantes que conocemos hasta el momento son `figure` y `table`, pero podemos definir nuevos entornos flotantes utilizando el paquete float. El contenido y el formato de los flotantes son definidos por el usuario y el comando `\caption` define el formato del título.

Estilo de flotante

Los estilos de los flotantes son la apariencia de cada clase de un flotador; por ejemplo, la clase de flotador `figure` tiene su propia apariencia. Con los estilos de los flotadores no nos estamos refiriendo a los estilos de una página, en donde tiene su propia apariencia de tipo `plain`, `empty` y `headings`. Los estilos de flotantes existentes son:

1. `plain`: este estilo de flotante no tiene ningún tipo de apariencia solo que su `caption` aparece debajo del objeto flotante.
2. `plaintop`: este estilo de flotante es igual al anterior pero el `caption` aparece arriba del objeto flotante.
3. `boxed`: el cuerpo del flotante se encierra en una caja y el `caption` aparece abajo de la caja.
4. `ruled`: es un estilo de tipo tabla en donde el cuerpo está rodeado por una línea por la parte superior e inferior y el `caption` aparece en la parte superior.

El paquete float proporciona el comando `\newfloat` para definir nuevos clases de flotantes, se debe tener en cuenta que no puede existir dos clases del mismo nombre. El comando `\listof` imprime la lista de flotantes de un tipo dado; por ejemplo, `\listoffigures` y `\listoftables` imprime una lista de figuras y tablas, respectivamente.

Comando newfloat

El comando `\newfloat` es similar al comando `\newtheorem`.

`\newfloat`

Sintaxis del comando `\newfloat`:

```
\newfloat{<clase or type>}{<posición o placement>}{<ext>}[<within>]
```

En donde,

1. `<clase or type>`: es el nombre de la clase de flotante.
2. `<posición o placement>`: es la posición en donde se colocará el flotante utilizando los valores como `t`, `b`, `p`, `h` o `H`.
3. `<ext>`: es la extensión del archivo que se generará para la lista de flotantes.
4. `<within>`: este argumento determina si las clases serán enumeradas dentro de una sección del documento LaTeX; por ejemplo, `<within>=chapter` serán enumeradas dentro del capítulo, caso que sucede con los flotantes `figure` y `table`.

```
\documentclass{article}
\usepackage[paperwidth=11cm,paperheight=4cm,margin=5mm]{geometry}
\usepackage{float}
\usepackage{minted}
\floatstyle{ruled} %Esta línea debe ir en preámbulo
\newfloat{Programa}{H}{pyt}[section] %Esta línea debe ir en preámbulo
\begin{document}
\section{Sección de prueba}
\begin{Programa}
  \begin{minted}{python}
    def suma(a,b):
      return a+b
  \end{minted}
  \caption{Programa en Python}
\end{Programa}
\end{document}
```

El estilo de un flotador determina la apariencia del flotador y del caption.

7.4 Objetos flotantes dentro del texto

Cuando se compila un documento LATEX el sistema TEXdivide el texto en parrafos alineados verticalmente.

Entorno wrapfigure y floatfigure

Para utilizar estos entornos es necesario importar los paquetes:

```
\usepackage{wrapfig}
\usepackage[rflt]{floatflt}
```

El paquete `wrapfig` proporciona el entorno `wrapfigure` que permite colocar un objeto flotante a la derecha o izquierda del texto (también proporciona el entorno `wraptable` para realizar tablas). El entorno `wrapfigure` tiene la siguiente sintaxis:

wrapfigure

```
\begin{wrapfigure}[<number of narrow
↪  lines>]{<placement>}[<overhang>]{<width>}
  <figure>
\end{wrapfigure}
```

Los valores del `<placement>` pueden ser:

1. `r`: a la derecha del texto.

2. `l`: a la izquierda del texto.
3. `i`: en borde interio, cerca de la encuadernación (funciona en documento de dos caras).
4. `o`: en borde externo, lejos de la encuadernación (funciona en documento de dos caras).

Ejemplo 7.6

```
\documentclass{article}
\usepackage[paperwidth=11cm,paperheight=10cm,margin=5mm]{geometry}
\usepackage{wrapfig,lipsum,graphicx}
\begin{document}
\section{Sección de prueba}
\begin{wrapfigure}{r}{5cm}
  \centering
  \includegraphics{images/grafica1}
  \caption{Gráfica 1}
  \label{fig:grafica1}
\end{wrapfigure}
\lipsum[1-2]
\end{document}
```

1 Sección de prueba

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in,

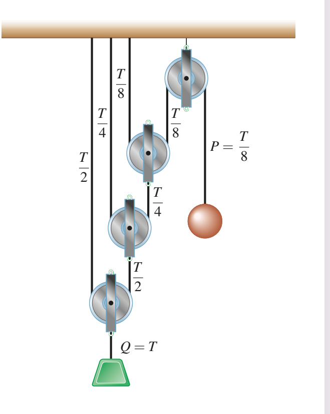


Figure 1: Gráfica 1

pretium quis, viverra ac, nunc.
Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Ejemplo 7.7

```
\documentclass{article}
\usepackage[paperwidth=11cm,paperheight=10cm,margin=5mm]{geometry}
\usepackage{wrapfig,lipsum,graphicx}
\begin{document}
\section{Sección de prueba}
\begin{wrapfigure}{l}{5cm}
```

```
  \centering
  \includegraphics{images/grafica1}
  \caption{Gráfica 1}
  \label{fig:grafica1}
\end{wrapfigure}
\lipsum[1-2]
\end{document}
```

1 Sección de prueba

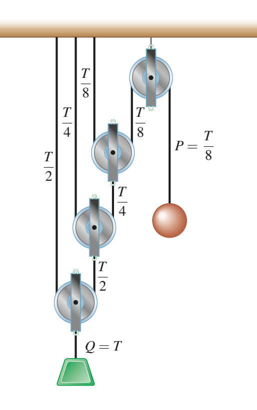


Figure 1: Gráfica 1

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

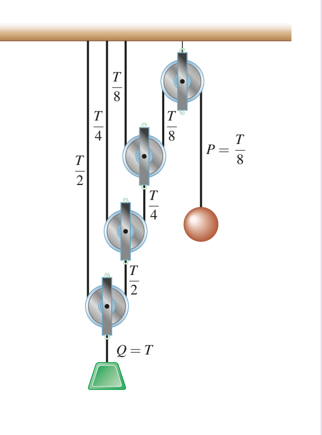
El entorno `floatingfigure` es similar al entorno `wrapfigure`. El argumento opcional del paquete especifica el posicionamiento de la figura. En este caso, `rflt` indica que las figuras y las tablas flotantes se muestren a la derecha de un párrafo. Con el argumento `lflt` las figuras y tablas se muestran a la izquierda de un párrafo y el valor `vflt` muestra las figuras y tablas a la derecha de un párrafo en las páginas impares y a la izquierda en las páginas pares.

Argumentos de floatfigure

Ejemplo 7.8

```
\documentclass{article}
\usepackage[paperwidth=11cm,paperheight=10cm,margin=5mm]{geometry}
\usepackage{lipsum,graphicx}
\usepackage[rflt]{floatflt}
\begin{document}
\begin{floatingfigure}{5cm}
  \centering
  \includegraphics{images/grafica1}
\end{floatingfigure}
\lipsum[1-2]
\end{document}
```

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.



Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

7.5 Leyendas flotantes

Paquete caption

Este paquete es utilizado para personalizar los subtitulo (caption) de los entornos flotantes como figure y table.

Uso del paquete

Para utilizar el paquete `caption` es necesario importar este paquete en el preámbulo del documento LaTeX.

```
\usepackage[<options>]{caption}
```

Las opciones proporcionadas al momento de importar el paquete también se pueden proporcionar por medio del comando.

```
\captionsetup[<float type>]{<options>}
```

El código

```
\usepackage[font=small,labelfont=bf,margin=5pt]{caption}
```

es equivalente a

```
\usepackage{caption}
\captionsetup{font=small,labelfont=bf,margin=5pt}
```

y se obtiene el mismo resultado.

El comando `captionsetup` permite personalizar el estilo de los subtitulos de los entornos flotantes y tiene efecto solo en el entorno donde se invoca. Entonces, para cambiar el estilo de los títulos de una figura o tabla debemos invocar el comando `captionsetup` justo antes del `caption`.

captionsetup

```
\begin{figure}
  ...
  \captionsetup{font=small,labelfont=bf,margin=5pt}
  \caption{...}
\end{figure}
```

Formateado

format

El título de una figura o tabla está constituido por el título (la que dice figura o tabla), un número (contador de figuras) y una descripción de la figura o tabla.

El formato del título de una figura o tabla se establece con la opción

```
format = <format name>
```

`plain` y `hang`: son los formatos ya definidos. Para declarar un nuevo formato existe el comando.

```
\DeclareCaptionFormat
```

Ejemplo 7.9

```
\documentclass{article}
\usepackage[paperwidth=11cm,paperheight=10cm,margin=5mm]{geometry}
\usepackage{lipsum,graphicx}
\begin{document}
\section{Sección de prueba}
\begin{figure}
  \centering
  \includegraphics[width=4cm]{images/grafica1}
  \caption{Descripción de la figura sin aplicar un formato}
\end{figure}
\lipsum[1-2]
\end{document}
```

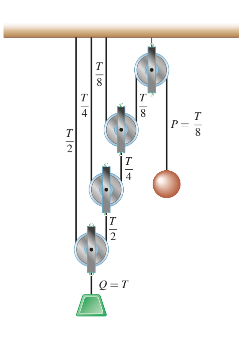


Figure 1: Descripción de la figura sin aplicar un formato

1 Sección de prueba

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Ejemplo 7.10

```
\documentclass{article}
\usepackage[paperwidth=11cm,paperheight=10cm,margin=5mm]{geometry}
\usepackage{lipsum,graphicx}
\usepackage{caption}
\begin{document}
\section{Sección de prueba}
\begin{figure}
  \centering
  \includegraphics[width=4cm]{images/grafica1}
  \captionsetup{format=hang}
  \caption{Descripción de la figura de formato hang. Este formato aplica sangria a descripción de la
  ↪  figura}
\end{figure}
\lipsum[1-2]
\end{document}
```

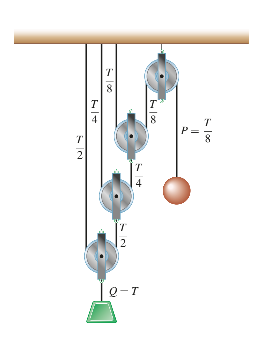


Figure 1: Descripción de la figura de formato hang. Este formato aplica sangria a descripción de la figura

1 Sección de prueba

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

La indentación de la descripción de la figura se puede controlar con la opción

indention

```
indention = <amount>
```

Ejemplo 7.11

```
\documentclass{article}
\usepackage[paperwidth=11cm,paperheight=10cm,margin=5mm]{geometry}
\usepackage{lipsum,graphicx}
\usepackage{caption}
\begin{document}
\section{Sección de prueba}
\begin{figure}
  \centering
  \includegraphics[width=4cm]{images/grafica1}
  \captionsetup{format=hang,indention=-0.8cm}
  \caption{Descripción de la figura de formato hang. Este formato aplica sangria a descripción de la
  ↪  figura}
\end{figure}
\lipsum[1-2]
\end{document}
```

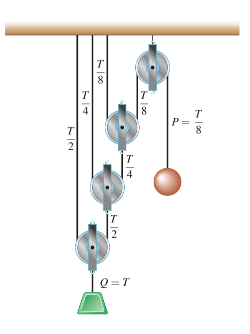


Figure 1: Descripción de la figura de formato hang. Este formato aplica sangria a descripción de la figura

1 Sección de prueba

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

labelformat

Para cambiar la etiqueta (formato de la enumeración) de la figura o tabla se puede recurir a la opción

```
labelformat = <label format name>
```

Las opciones definidos para este key son: `empty`, `simple`, `brace`, `parens` y para definir nuestros propios formatos se de realizar con el comando

```
\DeclareCaptionLabelFormat
```

Ejemplo 7.12

```
\documentclass{article}
\usepackage[paperwidth=11cm,paperheight=10cm,margin=5mm]{geometry}
\usepackage{lipsum,graphicx}
\usepackage{caption}
\begin{document}
\section{Sección de prueba}
\begin{figure}
  \centering
  \includegraphics[width=4cm]{images/grafica1}
  \captionsetup{labelformat=parens}
  \caption{Descripción de la figura sin enumeración}
\end{figure}
\lipsum[1-2]
\end{document}
```

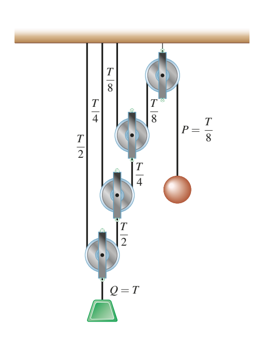


Figure (1): Descripción de la figura sin enumeración

1 Sección de prueba

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

El formato de la separación de la etiqueta (label) y la descripción de la figura o tabla se puede controlar con la opción

labelsep

```
labelsep = <label separator name>
```

Balores definidos para esta opción: `none`, `colon`, `period`, `space`, `quad`, `newline`. Para definir un nuevo valo de esta opción se debe recurrir al comando

```
\DeclareCaptionLabelSeparator
```

Ejemplo 7.13

```
\documentclass{article}
\usepackage[paperwidth=11cm,paperheight=10cm,margin=5mm]{geometry}
\usepackage{lipsum,graphicx}
\usepackage{caption}
\begin{document}
\section{Sección de prueba}
\begin{figure}
  \centering
  \includegraphics[width=4cm]{images/grafica1}
  \captionsetup{labelsep=newline}
  \caption{Descripción de la figura con espacio entre la etiqueta y la descripción}
\end{figure}
\lipsum[1]
\end{document}
```

1 Sección de prueba

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

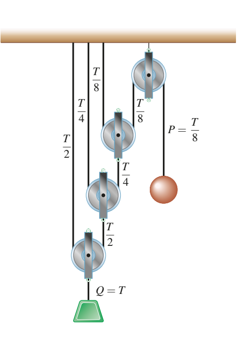


Figure 1
Descripción de la figura con espacio entre la etiqueta y la descripción

Justificaciones

justification

La justificación del título de una figura o tabla se puede controlar con la opción

```
justification = <justification name>
```

Valores establecidos para esta opción: `centering`, `centerlast`, `centerfirst`, `raggedright` y `raggedleft`.

Ejemplo 7.14

```
\documentclass{article}
\usepackage[paperwidth=11cm,paperheight=10cm,margin=5mm]{geometry}
\usepackage{lipsum,graphicx}
\usepackage{caption}
\begin{document}
\section{Sección de prueba}
\begin{figure}
  \centering
  \includegraphics[width=4cm]{images/grafica1}
  \captionsetup{justification=raggedright}
  \caption{Descripción de la figura con justificación a la izquierda}
\end{figure}
\lipsum[1-2]
\end{document}
```

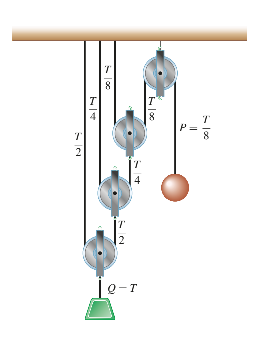


Figure 1: Descripción de la figura con justificación a la izquierda

1 Sección de prueba

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque.

Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Fuentes

Para controlar las fuentes del caption existen los siguientes opciones:

font, labelfont y textfont

```
font={<font options>}
labelfont={<font options>}
textfont={<font options>}
```

Valores para estas opciones: `scriptsize`, `footnotesize`, `small`, `normalsize`, `large`, `Large`, `normalfont`, `up`, `it`, `{Sl}`, `{SC}`, `md`, `bf`, `{rm}`, `Sf`, `tt`.

Ejemplo 7.15

```
\documentclass{article}
\usepackage[paperwidth=11cm,paperheight=10cm,margin=5mm]{geometry}
\usepackage{lipsum,graphicx}
\usepackage{caption}
\begin{document}
\section{Sección de prueba}
\begin{figure}
  \centering
  \includegraphics[width=4cm]{images/grafica1}
  \captionsetup{
    font={large},
    labelfont={bf,it},
    textfont={up}
  }
```

```
  \caption{Descripción de la figura}
\end{figure}
\lipsum[1-2]
\end{document}
```

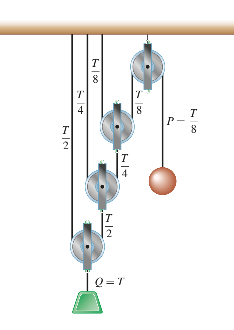


Figure 1: Descripción de la figura

1 Sección de prueba

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque.

Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Márgenes

margin y width

Opciones de márgenes del caption se pueden controlar con las siguientes opciones:

```
margin = <amount>
margin = {<left amount>, <right amount>}
width = <amount>
```

Ejemplo 7.16

```
\documentclass{article}
\usepackage[paperwidth=11cm,paperheight=10cm,margin=5mm]{geometry}
\usepackage{lipsum,graphicx}
\usepackage{caption}
\begin{document}
\section{Sección de prueba}
\begin{figure}
  \centering
  \includegraphics[width=4cm]{images/grafica1}
  \captionsetup{width=3cm}
  \caption{Descripción de la figura}
\end{figure}
```

```
\lipsum[1]
\end{document}
```

1 Sección de prueba

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

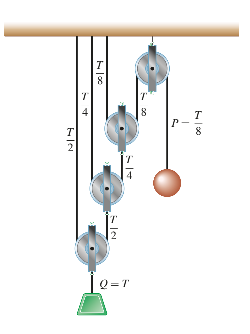


Figure 1: Descripción de la figura

Estilos

La configuración de las opciones se resumen a un estilo de caption como el estilo de página. La opción `style = <style name>` define el estilo del caption. base y default: son estilos definidos existentes que se pueden utilizar. Para definir un nuevo estilo se debe de recurrir al comando

style

```
\DeclareCaptionStyle
```

Posiciones

Para posicionar un caption se debe recurrir al opción

```
position = <position name>
```

Valores para esta opción: `top`, `above`, `bottom` y `below`

Ejemplo 7.17

```
\documentclass{article}
\usepackage[paperwidth=11cm,paperheight=10cm,margin=5mm]{geometry}
\usepackage{lipsum,graphicx}
\usepackage{caption}
\begin{document}
\section{Sección de prueba}
\begin{figure}
```

```
  \centering
  \includegraphics[width=4cm]{images/grafica1}
  \captionsetup{position=top}
  \caption{Descripción de la figura}
\end{figure}
\lipsum[1-2]
\end{document}
```

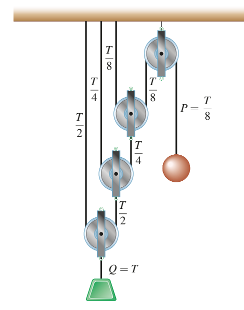


Figure 1: Descripción de la figura

1 Sección de prueba

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Listas

list

El comando `caption` genera una entrada en la lista de figuras o tablas, para suprimir este efecto debemos de recurrir a la opción

```
list = no
```

El formato de la lista se puede cambiar por medio de la opción.

```
listformat = <list format name>
```

`empty`, `simple`, `parens`, `subsimple` y `subparens`: son algunos valores de esta opción.

Nombres

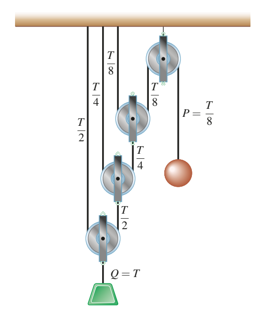
name

Los nombres como `figure` o `table` de la figura o tabla se pueden cambiar por medio de la opción

```
name = <name>
```

Ejemplo 7.18

```
\documentclass{article}
\usepackage[paperwidth=11cm,paperheight=10cm,margin=5mm]{geometry}
\usepackage{lipsum,graphicx}
\usepackage{caption}
\begin{document}
\section{Sección de prueba}
\begin{figure}
  \centering
  \includegraphics[width=4cm]{images/grafica1}
  \captionsetup{name=Gráfica}
  \caption{Descripción de la figura}
\end{figure}
\lipsum[1-2]
\end{document}
```



Gráfica 1: Descripción de la figura

1 Sección de prueba

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Tipos

Cuando se invoque el comando `caption` dentro de un entorno como figure o table, esto inmediatamente pone como nombre figure o table pero si nosotros invocamos dentro de un entorno cualquiera nos mandará un error al momento de compilar el documento latex, por tal motivo es necesario indicar el tipo de flontante.

type

```
type = <float type>
```

Configuraciones de opciones en el preámbulo

Para establecer las opciones de caption en el preámbulo de un documento LaTeXse debe recurrir al comando

```
\captionsetup[<float entry>]{<options>}
```

Ejemplo 7.19

```
\documentclass{article}
\usepackage{geometry}
\geometry{paperwidth=10cm, paperheight=8cm, margin=5mm}
\usepackage{graphicx}
\usepackage{caption}
\captionsetup[figure]{labelfont={bf}, name={Fig.}}
\begin{document}
\begin{figure}[h!]
  \centering
  \includegraphics[width=4cm]{images/grafica1}
  \caption{Descripción de la figura}
\end{figure}
\end{document}
```

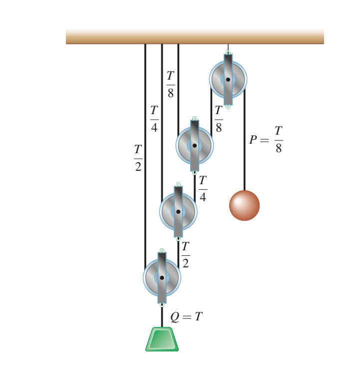


Fig. 1: Descripción de la figura

Declaración de nuevos formatos

\DeclareCaptionFormat

Para declarar un nuevo formato de caption se debe recurrir al comando

```
\DeclareCaptionFormat{<name>}{<code using #1, #2 and #3>}
```

En donde #1 es el nombre, #2 el número de la figura y #3 la descripción.

Ejemplo 7.20

```
\documentclass{article}
\usepackage{geometry}
\geometry{paperwidth=10cm, paperheight=8cm, margin=5mm}
\usepackage{graphicx,tikz}
\usepackage{caption}
\DeclareCaptionFormat{myformat}{#1#2#3\par}
\captionsetup[figure]{format=myformat}
\begin{document}
\begin{figure}[h!]
  \centering
  \includegraphics[width=4cm]{images/grafica1}
  \caption{Descripción de la figura}
\end{figure}
\end{document}
```

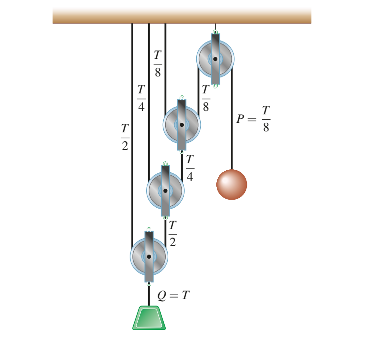


Figure 1: Descripción de la figura

Para declara un nuevo formato de etiqueta se debe recurrir al comando

`\DeclareCaptionLabelFormat`

```
\DeclareCaptionLabelFormat{<name>}{<code using #1 and #2>}
```

En donde #1 es el nombre y #2 el número de la figura.

Ejemplo 7.21

```
\documentclass{article}
\usepackage{geometry}
\geometry{paperwidth=10cm, paperheight=8cm, margin=5mm}
\usepackage{graphicx,tikz}
\usepackage{caption}
\DeclareCaptionLabelFormat{myformat}{\tikz[baseline]{\node[fill=red!50!white, rounded corners=3pt,
    anchor=base]{#1};}#2}
```

```
\captionsetup[figure]{labelformat=myformat}
\begin{document}
\begin{figure}[h!]
  \centering
  \includegraphics[width=4cm]{images/grafica1}
  \caption{Descripción de la figura}
\end{figure}
\end{document}
```

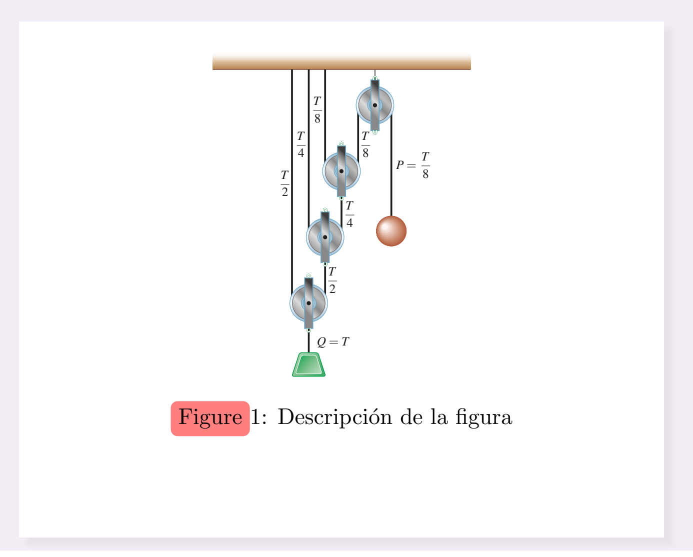


Figure 1: Descripción de la figura

`\DeclareCaptionTextFormat`

La declaración de un nuevo formato de texto se realiza por medio del comando

```
\DeclareCaptionTextFormat{<name>}{<code using #1>}
```

En donde #1 es el texto de la figura.

Ejemplo 7.22

```
\documentclass{article}
\usepackage{geometry}
\geometry{paperwidth=10cm, paperheight=8cm, margin=5mm}
\usepackage{graphicx,tikz}
\usepackage{caption}
\DeclareCaptionTextFormat{myformat}{\tikz[baseline]{\node[fill=red!50!white, rounded corners=3pt,
↪  anchor=base]{#1};}}
\captionsetup[figure]{textformat=myformat}
\begin{document}
\begin{figure}[h!]
  \centering
  \includegraphics[width=4cm]{images/grafica1}
  \caption{Descripción de la figura}
```

```
\end{figure}
\end{document}
```

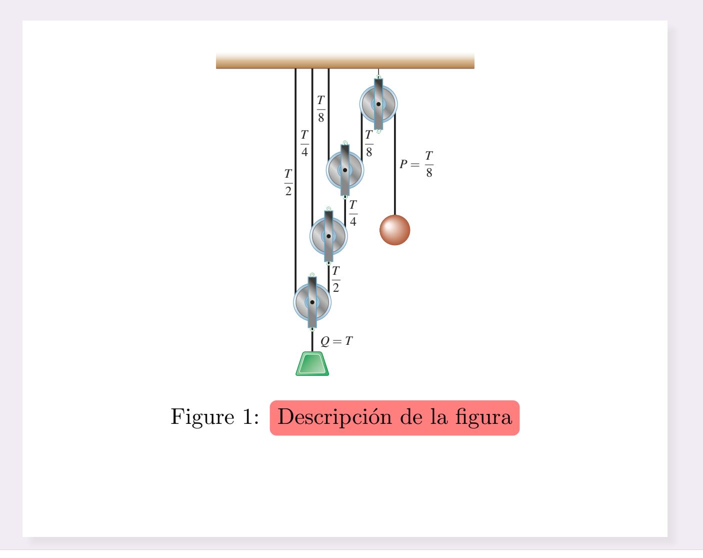


Figure 1: Descripción de la figura

El separador de la etiqueta y el texto se puede declarar por medio del comando

\DeclareCaptionLabelSeparator

```
\DeclareCaptionLabelSeparator{<name>}{<code>}
```

7.6 Paquete floatrow

Floatrow

Al momento de cargar este paquete todo el contenido de un entorno flotante (figure y table) se centra automáticamente por lo que es necesario invocar algún comando de alineación para obtener otra alineación del contenido del entorno flotante, y los títulos se sitúan debajo del contenido del entorno flotante.

Entorno flotante

Un entorno flotante es de la forma.

```
\begin{<float type>}
  <float content>
  \caption{<caption content>}
\end{<float type>}
```

\floatrowsetup

El comando `\floatsetup` permite configurar el comportamiento de los entornos flotantes.

```
\usepackage{floatrow}
\floatsetup[<float type>]{<options>}
```

Macros para crear nuevos objetos flotantes

\ttabbox

El comando `\ttabbox` construye el contenido del entorno tabla, en donde el título se ubica por encima de la tabla y es igual al ancho de la tabla.

Ejemplo 7.23

```
\documentclass{article}
\usepackage[paperwidth=11cm,paperheight=10cm,margin=5mm]{geometry}
\usepackage{lipsum,graphicx,floatrow}
\begin{document}
\section{Sección de prueba}
\begin{table}
  \ttabbox{%
    \caption{Descripción de la tabla}
  }{%
    \begin{tabular}{|cc|c|}
      \hline
      p   &   q   &   p $\wedge$ q\\
      \hline
      V   &   V   &   V\\
      V   &   F   &   F\\
      F   &   V   &   F\\
      F   &   F   &   F\\
      \hline
    \end{tabular}
  }
\end{table}
\lipsum[1]
\end{document}
```

Table 1: Descripción de la tabla

p	q	$p \wedge q$
V	V	V
V	F	F
F	V	F
F	F	F

1 Sección de prueba

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

`\ffigbox` Al igual que el comando `\ttabbox` el comando `\ffigbox` construye el contenido del entorno figura, en donde el título del contenido se ubica por debajo de la figura y es igual al ancho del texto.

Ejemplo 7.24

```
\documentclass{article}
\usepackage[paperwidth=11cm,paperheight=10cm,margin=5mm]{geometry}
\usepackage{lipsum,graphicx,floatrow}
\begin{document}
\section{Sección de prueba}
\begin{figure}
  \ffigbox{%
    \caption{Descripción de la figura}
  }{%
    \includegraphics[width=4cm]{images/grafica1}
  }
\end{figure}
\lipsum[1-2]
\end{document}
```

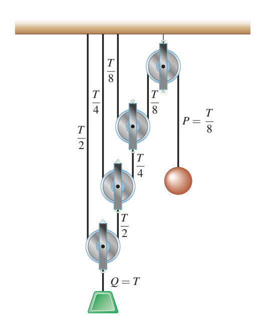


Figure 1: Descripción de la figura

1 Sección de prueba

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque.

Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Para que el título de una figura sea del mismo ancho que la figura se debe utilizar el comando `\FBwidth` como argumento opcional del comando `\ffigbox`.

`\FBwidth`

Ejemplo 7.25

```
\documentclass{article}
\usepackage[paperwidth=11cm,paperheight=10cm,margin=5mm]{geometry}
\usepackage{lipsum,graphicx,floatrow}
\begin{document}
\section{Sección de prueba}
\begin{figure}
  \ffigbox[\FBwidth]{%
```

```
    \caption{Descripción de la figura}
  }{%
    \includegraphics[width=4cm]{images/grafica1}
  }
\end{figure}
\lipsum[1-2]
\end{document}
```

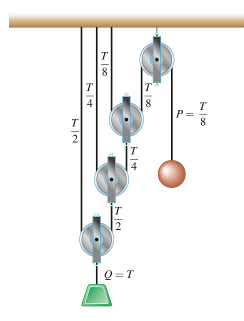


Figure 1: Descripción de la figura

1 Sección de prueba

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

`\fcapside`

El comando `\fcapside` coloca el título al lado de un entorno flotante. El título se ubica en el lado de encuadernación del documento LaTeX . El título y la figura son del mismo ancho por lo que el ancho de texto se divide en dos partes iguales.

Ejemplo 7.26

```
\documentclass{article}
\usepackage[paperwidth=11cm,paperheight=10cm,margin=5mm]{geometry}
\usepackage{lipsum,graphicx,floatrow}
\begin{document}
\section{Sección de prueba}
\begin{figure}
  \fcapside{%
    \caption{Descripción de la figura}
    \label{fig:ejemplofcapside}
  }{%
    \includegraphics[width=4cm]{images/grafica1}
  }
\end{figure}
```

```
\lipsum[1-2]
\end{document}
```

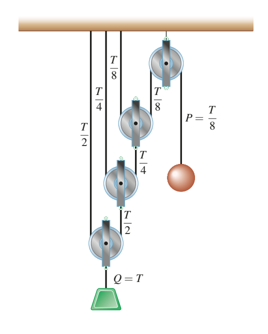


Figure 1: Descripción de la figura

1 Sección de prueba

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

La opción `\FBwidth` del comando `\fcapside` permite que la figura mantenga su ancho y el resto del ancho de texto es ocupado por el título de la figura.

`\FBwidth`

`floatrow`

Cajas flotantes en fila

Para colocar objetos flotantes uno al lado del otro se debe de recurrir al entorno `floatrow`.

Ejemplo 7.27

```
\documentclass{article}
\usepackage[paperwidth=11cm,paperheight=10cm,margin=5mm]{geometry}
\usepackage{lipsum,graphicx,floatrow}
\begin{document}
\section{Sección de prueba}
\begin{figure}
  \begin{floatrow}
    \ffigbox{\caption{Descripción de la figura}}{\includegraphics[width=4cm]{images/grafica1}}
    \ffigbox{\caption{Descripción de la figura}}{\includegraphics[width=4cm]{images/grafica1}}
  \end{floatrow}
\end{figure}
\lipsum[1-2]
\end{document}
```

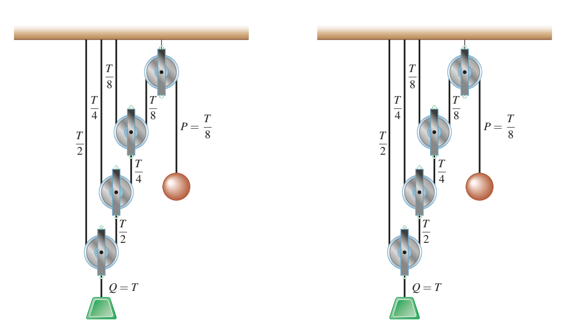


Figure 1: Descripción de la figura

Figure 2: Descripción de la figura

1 Sección de prueba

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

En vez de `\ffigbox` se puede utilizar `\ttabbox` para crear tablas en fila.

`\rawfloats`

Para restaurar el comportamiento estándar del LaTeX sobre objetos flotantes se debe invocar el comando `\rawfloats`. El comando mencionado suele utilizarse cuando se invocan los diferentes comandos proporcionados por el paquete `floatrow`.

Macros para objetos flotantes

Macro floatbox

`\floatbox`

Los comandos `\ttabbox`, `\ffigbox` y `\fcapside` han sido construidos utilizando el comando `\floatbox`. El comando `\floatbox` Tiene la capacidad de crear un cuadro flotante como `figure` o `table`. Sintaxis del comando `\floatbox`:

```
\floatbox[<preamble>]{<captype>}[<width>][<height>][<vert
↪ pos>]{<caption>}{<object>}
```

En donde,

`<preamble>` es un argumento que se encarga de posicionar el título del objeto flotante, sus valores son:

- `\capbeside`: Coloca el título al lado del objeto flotante.
- `\nocapbesid`: El título es colocado arriba o abajo dependiendo del estilo de flotante.
- `\captop`: El título es colocado arriba del objeto flotante.

`<captype>` es el tipo de objeto flotante que se desea crear.

`<width>` es el ancho del objeto flotante, si se deja vacío se utiliza el ancho natural del objeto.

`<height>` es la altura del objeto flotante, si se deja vacío o `[\hsize]` utiliza la altura natural del objeto.

`<vert pos>` es la posición vertical del objeto flotante, sus valores son:

- `t`: Alinea el objeto al borde superior.
- `c`: Alinea el objeto al centro.
- `b`: Alinea el objeto al borde inferior.
- `s`: Estira el objeto al máximo posible.

`<caption>` es el título del objeto flotante.

`<object>` es el contenido del objeto flotante.

`\FBwidth`, ancho de caption al del objeto

Si pasamos `\FBwidth` como argumento de `<width>`, el ancho del título será igual al ancho natural del objeto.

`\FBheight`

Nota: Si se utiliza el comando `\FBwidth` como argumento opcional de `<width>`, se debe de asegurar que el contenido del objeto pueda ser colocado en el comando `\hbox`. Sólo se permite utilizar el comando `\vspace`, no `\vskip`, al inicio y al final del contenido del objeto para ajustar el espacio vertical y la posición del contenido.

El comando `\FBheight` es un argumento opcional de `<height>`. Con fines prácticos se puede realizar operaciones como el cáculo de altura de objeto flotante y alterar sus valores como `[\FBheight+1cm]`.

Ejemplo 7.28

```
\documentclass{article}
\usepackage[paperwidth=10cm,paperheight=8cm,margin=5mm]{geometry}
\usepackage{caption,floatrow,float,graphicx}
\newcommand\rightlasat{\leftskip0pt plus 1 fil\rightskip0pt plus -1 fil\parfillskip0pt plus 1 fil}
\DeclareCaptionJustification{rightlast}{\rightlasat}
\begin{document}
\begin{figure}[H]
  \floatbox[{
    \capbeside
    \captionsetup[capbesidefigure]{labelsep=newline,justification=rightlast}%
    \thisfloatsetup{capbesideposition={left,center},capbesidewidth=4cm}
  }]{figure}[\FBwidth]
  {\caption{Descripción de la figura}\label{fig:1}}
  {\includegraphics[width=4cm]{images/grafica1}}
\end{figure}
\end{document}
```

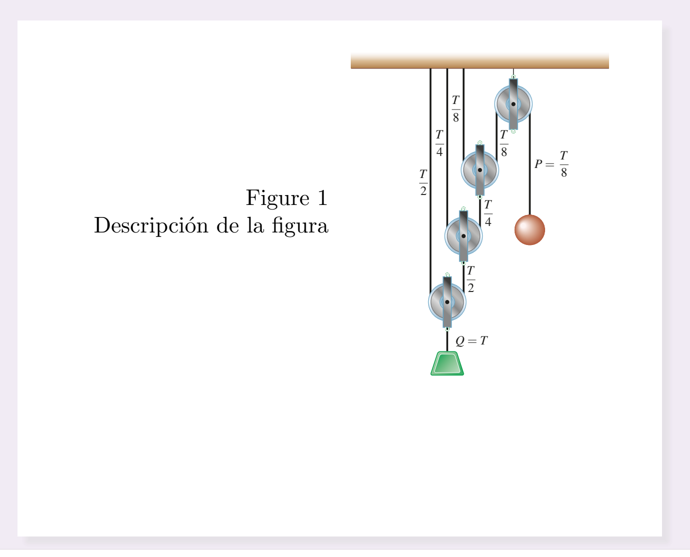


Figure 1
Descripción de la figura

Comandos personales para cajas flotantes

Utilizar el comando `\floatbox` sería aburrido y tedioso al momento de escribir un libro o artículo científico, por lo que es mejor crear nuevos comandos basados en el comando `\floatbox`.

`\newfloatcommand`

El comando `\newfloatcommand` es una macro que permite crear nuevos flotantes y su sintaxis es:

```
\newfloatcommand{<command>}{<captype>}[<preamble>][<default width>]
```

En donde,

`<command>` es el nombre del comando que se desea crear.

`<captype>` es el tipo de objeto flotante que se desea crear.

`<preamble>` es un argumento que se encarga de posicionar el título del objeto flotante, sus valores son:

- `\capbeside`: Coloca el título al lado del objeto flotante.
- `\nocapbesid`: El título es colocado arriba o abajo dependiendo del estilo de flotante.
- `\captop`: El título es colocado arriba del objeto flotante.

`<default width>` es el ancho por defecto del objeto flotante.

```
\newfloatcommand{fcapbesideleft}[{
  \capbeside
  \captionsetup[capbesidefigure]{labelsep=newline,justification=rightlast}%
  \thisfloatsetup{capbesideposition={left,center},capbesidewidth=4cm}
}][\FBwidth]
```

Formateado de objetos flotantes

Los objetos flotantes pueden ser formateados con el comando \floatsetup . La sintaxis del comando es:

\floatsetup

```
\floatsetup[&lt;float type&gt;]{&lt;options&gt;}
```

Por ejemplo, para cambiar el estilo del entorno figure a Plaintop se utiliza el comando:

```
\floatsetup[figure]{style=Plaintop}
```

Puntos importantes:

1. Se crea y cambia las opciones de los títulos de los entornos flotantes con el comando \captionsetup.
2. El comando \RawFloats cancela todas las opciones de formato de los entornos flotantes.
3. El entorno floatrow y los comandos \floatbox, \ffigbox y \ttabbox funcionan después de usar el comando \RawFloats.

Claves de configuraciones

style

Como estilo de flotante se puede personalizar los ajustes de justificación del contenido de flotante; los márgenes; la separación, entre objetos flotantes y los título.

El comando \DeclareFloatStyle permite crear nuevos estilos de flotantes.

```
\floatsetup[<float type>]{style=<style>}
```

La opción style permite cambiar el estilo de flotante, los valores que puede tomar son: plain, plaintop, Plaintop, ruled, Ruled, boxed, Boxed, BOXED y entre otros (ver la documentación oficial del paquete).

El clave capposition define la posición del título del objeto flotante. Valores para este clave:

capposition

TOP Coloca el título del objeto flotante arriba del objeto flotante y alineado por la parte superior.

beside Coloca el título del objeto flotante al lado del objeto flotante.

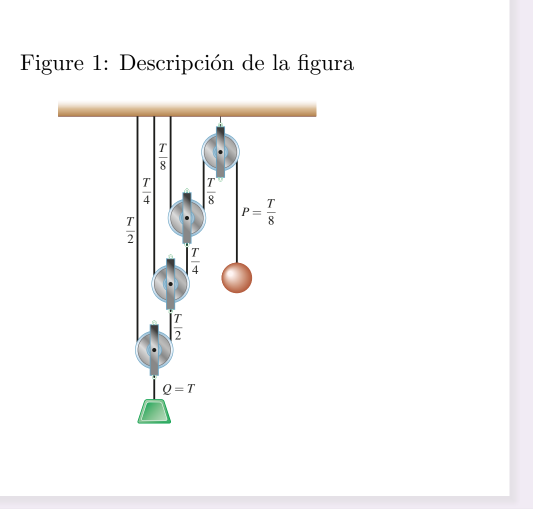
top Coloca el título del objeto flotante arriba del objeto flotante.

bottom Coloca el título del objeto flotante abajo del objeto flotante.

Ejemplo 7.29

```
\documentclass{article}
\usepackage[paperwidth=10cm,paperheight=8cm,margin=5mm]{geometry}
\usepackage{caption,floatrow,float,graphicx}
```

```
\floatsetup[figure]{style=plain,capposition=top}
\begin{document}
  \begin{figure}
    \includegraphics[width=4cm]{images/grafica1}
    \caption{Descripción de la figura}
  \end{figure}
\end{document}
```



capbesideposition

El clave `capbesideposition` define la posiciones al lado del objeto flotante. Existen dos grupos de valores para este clave: posicionamiento horizontal y posicionamiento vertical. Para posicionamiento horizontal existen cuatro valores:

left Coloca el título del objeto flotante a la izquierda del objeto flotante.

right Coloca el título del objeto flotante a la derecha del objeto flotante.

inside Coloca el título del objeto flotante en el lado de encuadernación de la página.

outside Coloca el título del objeto flotante en el lado exterior de la página.

Para el posicionamiento vertical existen tres valores:

top Coloca el título del objeto flotante arriba del objeto flotante.

bottom Coloca el título del objeto flotante abajo del objeto flotante.

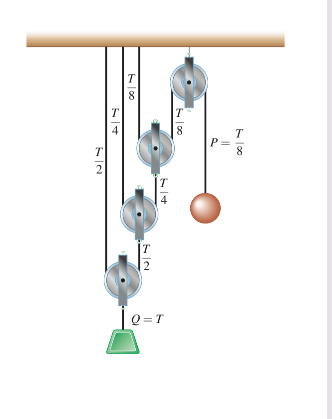
center Coloca el título del objeto flotante centrado en el objeto flotante.

Ejemplo 7.30

```
\documentclass{article}
\usepackage[paperwidth=10cm,paperheight=7cm,margin=5mm]{geometry}
```

```
\usepackage{caption,floatrow,float,graphicx}
\floatsetup[figure]{capposition=beside, capbesideposition={top,left}}
\begin{document}
  \begin{figure}
    \includegraphics[width=4cm]{images/grafica1}
    \caption{Descripción de la figura}
  \end{figure}
\end{document}
```

Figure 1: Descripción de la figura



El clave `capbesidewidth` define el ancho del título del objeto flotante. Si este clave no se define, el ancho del título del objeto flotante se calcula de acuerdo al ancho del objeto flotante.

capbesidewidth

```
\setupfloat[<float type>]{capbesidewidth=<width>}
```

floatwidth

El clave `floatwidth` define el ancho del objeto flotante. El valor por defecto de este clave es

```
floatwidth=\textwidth
```

Ejemplo 7.31

```
\documentclass{article}
\usepackage[paperwidth=10cm,paperheight=10cm,margin=5mm, showframe]{geometry}
\usepackage{caption,floatrow,float,graphicx}
\floatsetup[figure]{margins=raggedright, floatwidth=0.5\hsize}
\begin{document}
  \begin{figure}
    \includegraphics[width=\hsize]{images/grafica1}
    \caption{Descripción de la figura}
  \end{figure}
\end{document}
```

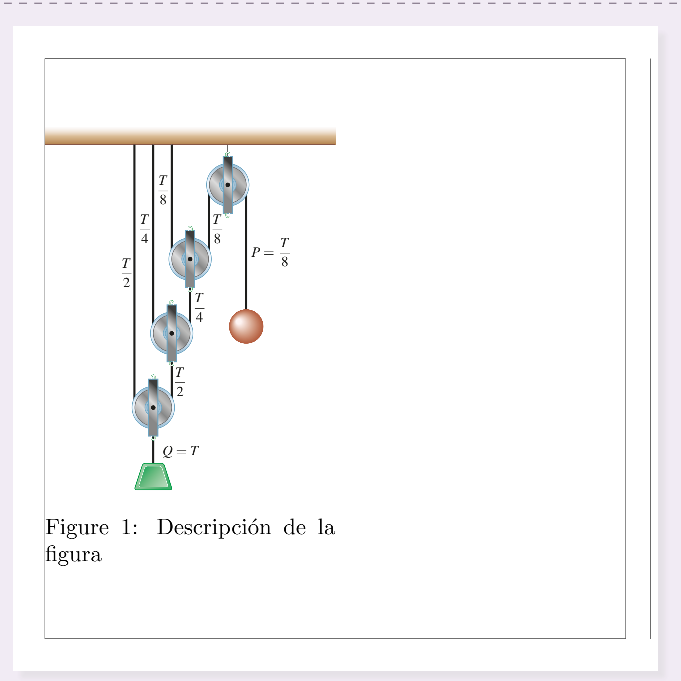


Figure 1: Descripción de la figura

Ejemplo 7.32

```
\documentclass{article}
\usepackage[paperwidth=10cm,paperheight=10cm,margin=5mm, showframe]{geometry}
\usepackage{caption,floatrow,float,graphicx}
\floatsetup[figure]{floatwidth=0.4\hsize, capbesidewidth=sidefil, capposition=beside,
↪  capbesideposition=right}
\begin{document}
  \begin{figure}
    \includegraphics[width=\hsize]{images/grafica1}
    \caption{Descripción de la figura}
  \end{figure}
\end{document}
```

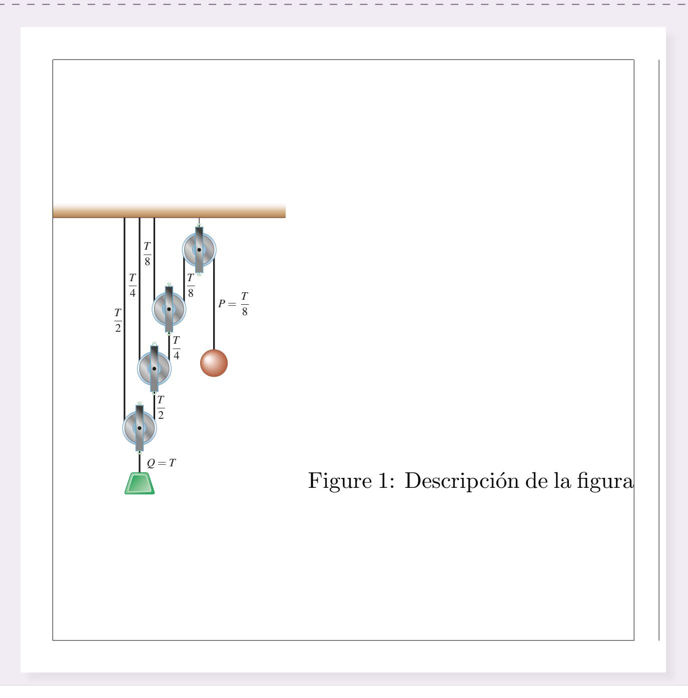


Figure 1: Descripción de la figura

La clave `facing` , si está establecido a `yes`, permite que el objeto flotante sea de diferente diseño dependiendo a las páginas pares e impares, siempre y cuando el tipo de documento LaTeX sea de 2 caras. Esta opción se utiliza generalmente para obtener el título de objeto flotante en el lado interno de la página del documento con la opción `capbesideposition=inside`.

facing

La clave `margins` define el margen del objeto flotante. Las opciones predefinidas son:

margins

centering Centra el objeto flotante.

raggedright Alinea el objeto flotante a la izquierda.

raggedleft Alinea el objeto flotante a la derecha.

hangleft Alinea el objeto flotante a la izquierda, pero el borde izquierdo del objeto flotante se alinea con el borde izquierdo de la página.

hangright Alinea el objeto flotante a la derecha, pero el borde derecho del objeto flotante se alinea con el borde derecho de la página.

hanginside Alinea el objeto flotante a la izquierda, pero el borde izquierdo del objeto flotante se alinea con el borde izquierdo de la página. Si el documento es de dos caras, el borde izquierdo del objeto flotante se alinea con el borde interno de la página.

hangoutside Alinea el objeto flotante a la derecha, pero el borde derecho del objeto flotante se alinea con el borde derecho de la página. Si el documento es de dos caras, el borde derecho del objeto flotante se alinea con el borde externo de la página.

floatrowsep

El clave `floatrowsep` define el espacio entre los las filas generados por `floatrow`.

capbesidesep

El clave capbesidesep define el espacio entre el objeto flotante y el título del objeto. Valores predefinidos para esta opción:

columnsep El espacio horizontal es de \columnsep y es valor por defecto de la clave.

quad El espacio horizontal es de 1em.

qquad El espacio horizontal es de 2em.

hfil El espacio horizontal es de 1fil.

hfill El espacio horizontal es de 1fill.

Aplicar opciones a un flontante actual

\thisfloatsetup

Para aplicar las opciones de configuración de flotación a un entorno flotante específico, se puede usar el comando \thisfloatsetup . Este comando tiene la misma sintaxis que \floatsetup. Este comando se debe de invocar justo antes del entorno flotante.

Borrado de opciones de un entorno flotante

\clearfloatsetup

Para limpiar las opciones aplicados a un entorno flotante específico, se puede usar el comando \clearfloatsetup . Sintaxis del comando \clearcaptionsetup:

```
\clearfloatsetup{<float type>}
```

Borrado temporal de todas opciones de un entorno flotante

\killfloatstyle

Para limpiar temporalmente todas las opciones aplicadas a un entorno flotante específico, se puede usar el comando \killfloatstyle. Este comando se debe de utilizar antes del comando \floatbox o justo después de invocar el entorno flotante.

```
\begin{figure}
  \killfloatstyle
  \floatsetup[<float type>]{<options>}
  ...
\end{figure}
```

Creación de nuevos entornos flotantes

\DeclareNewFloatType

Para crear un nuevo tipo de flotante este paquete nos proporciona el comando \DeclareNewFloatType . Sintaxis del comando \DeclareNewFloatType:

```
\DeclareNewFloatType{<type>}{<options>}
```

En donde, el `<type>` es el nombre del entorno flotante y `<options>` es una lista de opciones de tipo clave-valor (`<key>=<value>`). Las opciones que se pueden utilizar son las siguientes:

placement Define la posición del entorno flotante en la página. Puede tomar los valores `t`, `b`, `h` y `p`. Por defecto es `tbp`.

name Define el nombre del entorno flotante en el texto. Por defecto es el nombre del entorno flotante.

fileext Define la extensión del archivo en el que se almacena la lista de flotantes.

within Define la sección en la que se reinicia el contador de flotantes. Por defecto es la sección en la que se encuentra el entorno flotante. Puede adoptar los valores como el chapter y section.

Reemplazar newfloat por DeclareNewFloatType

El comando `\newfloat`

```
\newfloat{<type>}{<placement>}{<ext>}[<within>]
```

puede ser reemplazado por

```
\DeclareNewFloatType{<type>}{placement=<placement>, fileext=<ext>,
↪  within=<within>}
```

El paquete float también ofrece otros comandos para declarar nuevos tipos de flotantes. El comando `\floatname` puede ser reemplazado por la opción `name=key` del comando `\DeclareNewFloatType` y el comando `\floatplacement` puede ser reemplazado por la opción `placement` del comando `\DeclareNewFloatType`.

MODO MATEMÁTICO

8.1 Comandos más comunes

`^_`

Subíndices y superíndices

Ejemplo 8.1

```
Superíndice: $x^2$\\
Subíndice: $x_2$\\
Superíndice y subíndice: $x^2_3$
```

Superíndice: x^2
Subíndice: x_2
Superíndice y subíndice: x^2_3

`\over` y `\frac`

Fraciones

Las fracciones se pueden lograr con los comandos `\over` y `\frac`.

Ejemplo 8.2

```
$x+1 \over x-1$\\
$\frac{x+1}{x-1}$
```

$\frac{x+1}{x-1}$
$\frac{x+1}{x-1}$

Otros comandos

Ejemplo 8.3

```
Raices: $\sqrt{2}$.\\
Raices $n$-ésimas: $\sqrt[n]{2}$.\\
Integrales: $\int_{0}^{5} x^2 dx$.\\
Sumatoria: $\sum$.
```

Raices: $\sqrt{2}$.
Raices n-ésimas: $\sqrt[n]{2}$.
Integrales: $\int_{0}^{5} x^2 dx$.
Sumatoria: $\sum$.

8.2 Modo matemático en línea

El modo matemático en línea se puede lograr de tres maneras:

Ejemplo 8.4

```
\begin{enumerate}
  \item Primera forma: \(c^2=a^2+b^2\).
  \item Segunda forma: $c^2=a^2-b^2$.
  \item Tercera forma:
    \begin{math}
      c^2=a^2+b^2
    \end{math}
\end{enumerate}
```

1. Primera forma: $c^2=a^2+b^2$.
2. Segunda forma: $c^2=a^2-b^2$.
3. Tercera forma: $c^2=a^2+b^2$

El más utilizado es la segunda forma y no se puede utilizar dentro de un entorno matemático.

Límites

^_

Los límites se tratan como simples superíndices y subíndices en modo línea. Por ejemplo,

Ejemplo 8.5

```
...$\int_{1}^{\infty}\frac{1}{x^2}dx=1$...
```

...$\int_{1}^{\infty}\frac{1}{x^2}dx=1$...

Con el comando `\limits` esto se puede cambiar:

Ejemplo 8.6

```
...$\int\limits_{1}^{\infty}{1}{x^2}dx=1$...
```

...$\int\limits_{1}^{\infty}1x^2dx=1$...

El comando `\limits` debe ir después del comando `\int` y antes de los límites.

Fracciones

Ejemplo 8.7

```
Fracciones en linea: $\frac{a}{b}$.
```

Fracciones en linea: $\frac{a}{b}$.

Numeración de ecuaciones

No se puede realizar las enumeraciones en modo en línea, además no tiene sentido realizar enumeraciones de ecuaciones en línea.

Expresiones matemáticas en cajas

\fbox

Para enmarcar una expresión matemática en una caja se puede utilizar el comando \fbox.

Ejemplo 8.8

```
...\fbox{$c^2=a^2+b^2$}...
```

... $\boxed{c^2=a^2+b^2}$ ...

Utilizando parámetros de \fbox:

Ejemplo 8.9

```
\fboxsep=5pt
\fboxrule=1pt
...\fbox{$c^2=a^2+b^2$}...
```

... $\boxed{c^2=a^2+b^2}$ ...

Utilizando el comando \colorbox se puede cambiar el color de la caja:

Ejemplo 8.10

```
...\colorbox{yellow}{$c^2=a^2+b^2$}...
```

... $c^2=a^2+b^2$ ...

\colorbox

Salto de línea

En LaTeX solo se puede separar las expresiones matemáticas cuando hay símbolos relacionales como =, < y >, y también se puede separar cuándo existe operadores binarios como + y −. Una expresión de tipo

```
$a+b+c$
```

se puede separar en dos o tres líneas, pero la expresión

```
${a+b+c}$
```

no se puede separar en dos o tres líneas, esto se debe a los dilimitadores.

Ejemplo 8.11

```
Expresion de varias líneas: $f(x)=a_{n} x^{n}+a_{n-1}
↪ x^{n-1}+a_{n-2} x^{n-2}+\ldots+a_{i} x^{i}+a_{2}
↪ x^{2}+a_{1} x^{1}+a_{0}$.\\
Expresion agrupado que no se puede separar: ${f(x)=a_{n}
↪ x^{n}+a_{n-1} x^{n-1}+a_{n-2} x^{n-2}+\ldots+a_{i}
↪ x^{i}+a_{2} x^{2}+a_{1} x^{1}+a_{0}}$
```

Expresion de varias líneas: $f(x)=a_{n} x^{n}+a_{n-1} x^{n-1}+a_{n-2} x^{n-2}+\ldots+a_{i} x^{i}+a_{2} x^{2}+a_{1} x^{1}+a_{0}$.
Expresion agrupado que no se puede separar: ${f(x)=a_{n} x^{n}+a_{n-1} x^{n-1}+a_{n-2} x^{n-2}+\ldots+a_{i} x^{i}+a_{2} x$

\mathsurround

Espacio en blanco antes y después de una expresión matemática

Para modificar el espacio ántes y después de la expresión se debe modificar la longitud \mathsurround.

Ejemplo 8.12

```
Sin modificar la longitud: $c^2=a^2+b^2$.\\
\setlength{\mathsurround}{15pt}
Modificado la longitud: $c^2=a^2+b^2$.
```

Sin modificar la longitud: $c^2 = a^2+b^2$.
Modificado la longitud: $c^2 = a^2+b^2$.

\displaystyle

8.3 Tamaño natural de expresiones matemáticas o displaymath

Las expresiones matemáticas en línea se adaptan al tamaño de la fuente, pero en tamaño natural esto no sucede. Par ingresar a tamaño natural se debe invocar el comando \displaystyle. Por ejemplo,

Ejemplo 8.13

```
Expresión metemática normal: $\frac{a}{b}$.\\
Expresión metemática en tamaño natural: $\displaystyle\frac{a}{b}$.
```

Expresión metemática normal: $\frac{a}{b}$.
Expresión metemática en tamaño natural: $\displaystyle\frac{a}{b}$.

Si se desea que solo una expresión pequeña (fragmento de código) esté en tamaño natural se debe utilizar el comando \displaystyle{}.

Las expresiones matemáticas dentro de entornos matemáticos ya están establecidas en tamaño natural.

Ecuaciones matemáticas

Para realizar las ecuaciones matemáticas en LaTeX existen varios entornos como equation, eqnarray, align y entre otros.

equation

Entorno equation

Ejemplo 8.14

```
\begin{equation}
  x + \frac{abc}{d} = 10
\end{equation}
```

$$x+\frac{abc}{d} = 10 \tag{1}$$

No existe el entorno \equation*.

Para obtener una ecuación sin enumeración podemos utilizar el entorno displaymath que es equivalente a \[\].

Ejemplo 8.15

```
\begin{displaymath}
  x + \frac{abc}{d} = 10
\end{displaymath}
```

$$x+\frac{abc}{d}=10$$

Ejemplo 8.16

```
\[x + \frac{abc}{d} = 10\]
```

$$x+\frac{abc}{d}=10$$

La enumeración de las ecuaciones o etiquétas en el entorno `equation` podemos quitarlo utilizando el comando `\nonumber`. Este comando también es utilizable para otros entornos con enumeración de ecuaciones.

`\nonumber`

Ejemplo 8.17

```
\begin{equation}
  x + \frac{abc}{d} = 10 \nonumber
\end{equation}
```

$$x+\frac{abc}{d}=10$$

Entorno eqnarray

El entorno eqnarray genera una matriz de 3 columnas y las filas que deseemos. Es como un array de columnas rcl (right, center, left). Para cambiar el comportamiento de este entorno es necesario modificar el entorno completo en el archivo latex.ltx.

Ejemplo 8.18

```
\begin{eqnarray*}
  \text{right}  & \text{center} & \text{left}\\
  \frac{1}{\sqrt{x}}  &= \frac{\sqrt{x}}{x} &= \frac{x}{x\sqrt{x}}
\end{eqnarray*}
```

$$\begin{array}{rcl} \text{right} & \text{center} & \text{left} \\ \frac{1}{\sqrt{x}} & =\frac{\sqrt{x}}{x} & =\frac{x}{x\sqrt{x}} \end{array}$$

Este entorno no se debe de sustituir por el entorno array.

Ejemplo 8.19

```
\begin{eqnarray}
  x + \frac{abc}{d} &= & 10\\
  x &= & 10 - \frac{abc}{d} \label{eq:1}
\end{eqnarray}
Ecuación \ref{eq:1}.
```

$$x+\frac{abc}{d} = 10 \quad (2)$$

$$x = 10-\frac{abc}{d} \quad (3)$$

Ecuación 5.

También es posible eliminar las enumeraciones de las ecuaciones en el entorno `eqnarray*` utilizando el comando `\nonumber`.

Ejemplo 8.20

```
\begin{eqnarray}
  x + \frac{abc}{d} &= & 10 \nonumber\\
  x &= & 10 - \frac{abc}{d}
\end{eqnarray}
```

$$x+\frac{abc}{d} = 10$$

$$x = 10-\frac{abc}{d} \quad (4)$$

Numeración de ecuaciones

Para eliminar las numeraciones de las ecuaciones se debe de utilizar los entornos con estrella.

Ejemplo 8.21

```
\begin{equation*}
  x + \frac{abc}{d} = 10
\end{equation*}
\begin{equation}
  x + \frac{abc}{d} = 10 \nonumber
\end{equation}
\begin{equation*}
  x + \frac{abc}{d} = 10 \nonumber
\end{equation*}
```

$$x+\frac{abc}{d}=10$$

$$x+\frac{abc}{d}=10$$

$$x+\frac{abc}{d}=10$$

Cambiar el estilo de las enumeraciones

Ejemplo 8.22

```
\documentclass{article}
\usepackage[paperwidth=10cm,paperheight=5cm,margin=2mm]{geometry}
\usepackage{amsmath}
\renewcommand\theequation{\textbf{Eq.\arabic{equation}}}
\begin{document}
  \begin{equation}
    a^2 + b^2 = c^2
  \end{equation}
  \section{Sección de prueba}
  \begin{equation}
    x^2 + \sqrt{abc} = 10
  \end{equation}
  \section{Segunda sección de prueba}
  \begin{equation}
    x^2 + y^2 = 10
  \end{equation}
\end{document}
```

$$a^2 + b^2 = c^2 \qquad (\textbf{Eq.1})$$

1 Sección de prueba

$$x^2 + \sqrt{abc} = 10 \qquad (\textbf{Eq.2})$$

2 Segunda sección de prueba

$$x^2 + y^2 = 10 \qquad (\textbf{Eq.3})$$

Cambiar la enumeración de las ecuaciones al izquierda

Para poner las enumeraciones a la izquierda de una ecuación matemática debemos de utilizar el paquete `leqno`.

leqno

Ejemplo 8.23

```
\documentclass{article}
\usepackage[paperwidth=10cm,paperheight=5cm,margin=2mm]{geometry}
\usepackage[leqno]{amsmath}
\makeatletter
\newcommand{\leqnomode}{\tagsleft@true}
\newcommand{\reqnomode}{\tagsleft@false}
\makeatother
```

```
\begin{document}
  \begin{align}
    f(x) &= ax^2 + bx + c \\
    g(x) &= dx^2 + ex + f
  \end{align}
  \reqnomode
  \begin{align}
    f(x) &= ax^2 + bx + c \\
    g(x) &= dx^2 + ex + f
  \end{align}
\end{document}
```

(1) $f(x) = ax^2 + bx + c$

(2) $g(x) = dx^2 + ex + f$

$f(x) = ax^2 + bx + c$ (3)

$g(x) = dx^2 + ex + f$ (4)

Etiquetas de ecuaciones

\label

Cualquier ecuación enumerada puede tener una etiqueta (label) y hacer una referencia a ella. Para hacer esto se debe utilizar el comando \label. Los nombres de las etiquetas no pueden contener caracteres de comandos de LaTeX. Un nombre de la etiqueta se reemplaza por el número de la ecuación. Por ejemplo:

Ejemplo 8.24

```
\begin{equation}
  x + \frac{abc}{d} = 10 \label{eq:1}
\end{equation}
Ecuación \ref{eq:1}.
```

$$x+\frac{abc}{d}=10 \tag{5}$$

Ecuación 5.

\tag

El comando \tag no permite hacer referencia a una etiqueta, pues simplemente asigna un nombre al ecuación.

Ejemplo 8.25

```
\begin{equation}
  x + \frac{abc}{d} = 10 \tag{eq:1}
\end{equation}
Ecuación \ref{eq:1}.
```

$$x+\frac{abc}{d}=10 \tag{eq:1}$$

Ecuación 5.

Marcos

El comando \fbox también se puede utilizar para enmarcar o poner en un cuadro las ecuaciones matemáticas o expresiones matemáticas, como se realiza en modo línea. No solamente podemos utilizar el comando \fbox, también podemos utilizar el comando \colorbox

Ejemplo 8.26

```
\fbox{\parbox{\linewidth-2\fboxrule-2\fboxsep}{%
    \begin{equation}
      x + \frac{abc}{d} = 10
    \end{equation}%
  }%
}
```

$$x + \frac{abc}{d} = 10 \qquad (6)$$

Si deseamos que el número de la ecuación no esté dentro marco, pues en ese caso se vuelve un poco complicado, pero podemos realizarlo esto de una manera más sencilla utilizando el paquete empheq y tcolorbox.

array

8.4 Arrays

Para realizar matrices en LaTeX debemos de recurrir al entorno array. Este entorno se comporta de manera similar que el entorno eqnarray, pero solo que en este caso hay la posibilidad de establecer el número de filas y columnas y además un array tiene solamente una enumeración de ecuación.

Ejemplo 8.27

```
\begin{equation}
  \left\{%
    \begin{array}{ccc}
      x & = & 10 \\
      y & = & 20
    \end{array}%
  \right.
\end{equation}
```

$$\left\{ \begin{array}{ccc} x & = & 10 \\ y & = & 20 \end{array} \right. \qquad (7)$$

El entorno array necesariamente tiene que estar dentro de un entorno matemático, con @{} antes de las primeras columnas y al final de la última columna.

Ejemplo 8.28

```
\begin{displaymath}
  \left\{%
    \begin{array}{@{\quad}ccc}
      x & = & 10 \\
      y & = & 20
    \end{array}%
  \right.
\end{displaymath}
```

$$\left\{ \begin{array}{@{\quad}ccc} x & = & 10 \\ y & = & 20 \end{array} \right.$$

La alineación horizontal se interpreta de manera similar que el entorno `tabular`.

Casos

Ejemplo 8.29

```
\begin{displaymath}
  \left|x\right| = \left\{%
    \begin{array}{ccc}
      x  &\text{si} & x \geq 0\\
      -x &\text{si} & x < 0
    \end{array}%
  \right.
\end{displaymath}
```

$$\left|x\right| = \left\{ \begin{array}{ccc} x & \text{si} & x \geq 0 \\ -x & \text{si} & x < 0 \end{array} \right.$$

arraycolsep

arraycolsep

La separación entre las columnas de un entorno `array` se las específica con el comando `arraycolsep`. La longitud `arraycolsep` en la mayoría de las clases está especificado en 5 puntos.

Sin modificar la longitud `arraycolsep`:

Ejemplo 8.30

```
\begin{displaymath}
  \left\{%
    \begin{array}{ccc}
      x & = & 10 \\
      y & = & 20
    \end{array}%
  \right.
\end{displaymath}
```

$$\left\{ \begin{array}{ccc} x & = & 10 \\ y & = & 20 \end{array} \right.$$

Modificando la longitud `arraycolsep`:

Ejemplo 8.31

```
\begingroup
\arraycolsep=1.4pt%
\begin{displaymath}
  \left\{%
    \begin{array}{ccc}
      x & = & 10 \\
      y & = & 20
    \end{array}%
  \right.
\end{displaymath}
\endgroup
```

$$\left\{\begin{array}{ccc} x & = & 10 \\ y & = & 20 \end{array}\right.$$

La modificación del longitud `arraycolsep` también afecta al entorno `eqnarray`.

Ejemplo 8.32

```
\begin{eqnarray*}
  x^2 + y^2 &= 25
\end{eqnarray*}
\begingroup
\arraycolsep=1.4pt%
\begin{eqnarray*}
  x^2 + y^2 &= 25
\end{eqnarray*}
\endgroup
```

$$x^2+y^2 \quad = 25$$

$$x^2+y^2 = 25$$

8.5 Matrices

Ejemplo 8.33

```
\begin{displaymath}
  \begin{matrix}
    x & y & z \\
    m & n & q \\
    r & s & t
  \end{matrix}
\end{displaymath}
```

$$\begin{matrix} x & y & z \\ m & n & q \\ r & s & t \end{matrix}$$

Ejemplo 8.34

```
\begin{displaymath}
  \begin{pmatrix}
    x & y & z \\
    m & n & q \\
    r & s & t
  \end{pmatrix}
\end{displaymath}
```

$$\begin{pmatrix} x & y & z \\ m & n & q \\ r & s & t \end{pmatrix}$$

Ejemplo 8.35

```
\begin{displaymath}
  \begin{bmatrix}
    x & y & z \\
    m & n & q \\
    r & s & t
  \end{bmatrix}
\end{displaymath}
```

$$\begin{bmatrix} x & y & z \\ m & n & q \\ r & s & t \end{bmatrix}$$

Ejemplo 8.36

```
\begin{displaymath}
  \begin{vmatrix}
    x & y & z \\
    m & n & q \\
    r & s & t
  \end{vmatrix}
\end{displaymath}
```

$$\begin{vmatrix} x & y & z \\ m & n & q \\ r & s & t \end{vmatrix}$$

Ejemplo 8.37

```
\begin{displaymath}
  \begin{Vmatrix}
    x & y & z \\
    m & n & q \\
    r & s & t
  \end{Vmatrix}
\end{displaymath}
```

$$\begin{Vmatrix} x & y & z \\ m & n & q \\ r & s & t \end{Vmatrix}$$

Ejemplo 8.38

```
\begin{displaymath}
  \begin{Bmatrix}
    x & y & z \\
    m & n & q \\
    r & s & t
  \end{Bmatrix}
\end{displaymath}
```

$$\begin{Bmatrix} x & y & z \\ m & n & q \\ r & s & t \end{Bmatrix}$$

Comando `\bordermatrix`:

Ejemplo 8.39

```
\begin{displaymath}
  \bordermatrix{%
      & 0 & 1 & 2 \cr
    0 & x & y & z \cr
    1 & m & n & q \cr
    2 & r & s & t \cr
  }
\end{displaymath}
```

$$\bordermatrix{ & 0 & 1 & 2 \cr 0 & x & y & z \cr 1 & m & n & q \cr 2 & r & s & t \cr }$$

8.6 Superíndices y subíndices

Al momento de escribir subíndices se genera un problema cuando queremos obtener en modo vertical. Por ejemplo,

Ejemplo 8.40

```
$V_{agua}$
```

V_{agua}

Podemos volver las letras de los subíndices en vertical con el comando `\text`:

Ejemplo 8.41

```
$V_{\text{agua}}$
```

V_{agua}

En este último caso, podemos apreciar que la altura de las letras no es buena, por tanto, no sería una solución efectiva. Podemos modificar los subíndices de la siguiente forma:

Ejemplo 8.42

```
$V_{\mbox{\vphantom{i}agua}}$
```

$V_{\mbox{\vphantom{i}agua}}$

Ejemplo 8.43

```
$V_{\mathrm{agua}}$
```

V_{agua}

8.7 Límites

Para los límites de las sumatorias y productorias se puede utilizar el comando `\atop`, pero este comando es como una fracción que no tiene una línea; por lo tanto, se recomienda utilizar el comando `\limits`.

Ejemplo 8.44

```
\begin{displaymath}
  \sum\limits_{i=1}^n x_i
\end{displaymath}
```

$$\sum\limits_{i=1}^n x_i$$

8.8 Raices

Para obtener las raíces se debe de utilizar el comando `\sqrt`.

Ejemplo 8.45

```
\begin{displaymath}
  \sqrt{x}
\end{displaymath}
```

$$\sqrt{x}$$

Para la raíz n-ésima:

Ejemplo 8.46

```
\begin{displaymath}
  \sqrt[n]{x}
\end{displaymath}
```

$$\sqrt[n]{x}$$

8.9 Delimitadores

Al utilizar los delimitadores, `()`, `[]`, `\{\}`, se genera un problema de los tamaños; por ejemplo,

Ejemplo 8.47

```
\begin{displaymath}
  E = (\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{x}}}}}})
\end{displaymath}
```

$$E = (\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{x}}}}}})$$

Para eliminar este problema debemos de recurrir a los comandos `\left` y `\right`. Después de los comandos mencionados debemos de proseguir con los delimitadores mencionados.

Ejemplo 8.48

```
\begin{displaymath}
  E = \left(\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{x}}}}}}\right)
\end{displaymath}
```

$$E = \left(\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{x}}}}}}\right)$$

Ejemplo 8.49

```
\begin{displaymath}
  E = \left[\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{x}}}}}}\right]
\end{displaymath}
```

$$E = \left[\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{x}}}}}}\right]$$

Ejemplo 8.50

```
\begin{displaymath}
  E = \left\{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{x}}}}}}\right\}
\end{displaymath}
```

$$E = \left\{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{x}}}}}}\right\}$$

Ejemplo 8.51

```
\begin{displaymath}
  E = \left|\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{x}}}}}}\right|
\end{displaymath}
```

$$E = \left|\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{x}}}}}}\right|$$

Ejemplo 8.52

```
\begin{displaymath}
  E = \left\|\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{x}}}}}}\right\|
\end{displaymath}
```

$$E = \left\|\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{\sqrt{x}}}}}}\right\|$$

Ejemplo 8.53

```
\begin{displaymath}
  \left\lfloor a^{2^{2^{2}}} + b^{2^{2^{2}}}\right\rfloor
\end{displaymath}
```

$$\left\lfloor a^{2^{2^{2}}} + b^{2^{2^{2}}}\right\rfloor$$

Para especificar tamaños o alturas fijas de los limitadores debemos de recurrir a los comandos `\big`, `\Big`, `\bigg` y `\Bigg`. Después de estos comandos mencionados debemos proseguirlos con los delimitadores.

Ejemplo 8.54

```
\begin{displaymath}
  \big(x\big)
\end{displaymath}
```

$$\big(x\big)$$

Ejemplo 8.55

```
\begin{displaymath}
  \bigg(x\bigg)
\end{displaymath}
```

$$\bigg(x\bigg)$$

Ejemplo 8.56

```
\begin{displaymath}
  \Big(x\Big)
\end{displaymath}
```

$$\Big(x\Big)$$

Ejemplo 8.57

```
\begin{displaymath}
  \Bigg(x\Bigg)
\end{displaymath}
```

$$\Bigg(x\Bigg)$$

Ejemplo 8.58

```
\begin{displaymath}
  A = \left\{x^{2^{2^{2}}}\Big/ x \geq 1\right\}
\end{displaymath}
```

$$A = \left\{x^{2^{2^{2}}}\Big/ x \geqslant 1\right\}$$

Nuevos delimitadores

Para declarar nuevos delimitadores debemos de utilizar el comando `\DeclareMathDelimiter`. Si se quiere modificar los delimitadores ya existentes, debemos de modificar el archivo fontmath.ltx en el directorio `texmf/tex/latex/base/fontmath.ltx`.

8.10 Texto en modo matemático

Para obtener textos verticales en modo matemático debemos de recurrir a los comandos `\mathrm`, `\textrm`, `\mbox` y `\text`.

Ejemplo 8.59

```
\begin{displaymath}
  \mathrm{texto en modo matemático}
\end{displaymath}
```

$$\mathrm{textoenmodomatemtico}$$

Ejemplo 8.60

```
\begin{displaymath}
  \textrm{texto en modo matemático}
\end{displaymath}
```

texto en modo matemático

Ejemplo 8.61

```
\begin{displaymath}
  \mbox{texto en modo matemático}
\end{displaymath}
```

texto en modo matemático

Ejemplo 8.62

```
\begin{displaymath}
  \text{texto en modo matemático}
\end{displaymath}
```

texto en modo matemático

8.11 Estilos de fuentes

El estilo de las fuentes de las expresiones matemáticas podemos cambiarlo de 2 maneras. La primera es utilizando el comando `\XX texto` y el segundo es utilizando el comando `\mathXX`. El XX de los comandos debemos reemplazar por estilos de los fuentes.

Ejemplo 8.63

```
\begin{displaymath}
  {\rm c^2 = a^2 + b^2}
\end{displaymath}
```

$$\mathrm{c^2 = a^2 + b^2}$$

Ejemplo 8.64

```
\begin{displaymath}
  \mathrm{c^2 = a^2 + b^2}
\end{displaymath}
```

$$\mathrm{c^2 = a^2 + b^2}$$

8.12 Espacios

Espacios horizontales

En LaTeXpor defecto no se deja espacios, pero hay modos para dejar espacios. Para dejar espacios en una expresión debemos de utilizar los comandos `\,`, `\;`, `\!`, `\:` y como la misma forma `\hhspace`.

Ejemplo 8.65

```
$x\in \mathbb{R}, n\in\mathbb{N}$
```

$x\in \mathbb{R}, n\in\mathbb{N}$

Ejemplo 8.66

```
$x\in\mathbb{R}, \; x\in\mathbb{N}$
```

$x\in\mathbb{R}, \; x\in\mathbb{N}$

Ejemplo 8.67

```
$a \text{ \textvisiblespace } b$
```

a ␣ b

Ejemplo 8.68

```
$a\quad b$
```

$a\quad b$

Ejemplo 8.69

```
$a \hspace{0.5cm} b$
```

$a \hspace{0.5cm} b$

Ejemplo 8.70

```
$a\kern0.5cm b$
```

$a\kern0.5cm b$

`\thinmuskip` `\medmuskip` `\thickmuskip`

En LaTeX se define tres longitudes:

```
\thinmuskip = 3mu
\medmuskip = 4mu plus 2mu minus 4mu
\thickmuskip = 5mu plus 5mu
```

Podemos modificar estos longitudes a nuestro gusta. Podemos realizar las modificaciones en el preámbulo del documento LaTeX para que se modifique globalmente.

Sea:

Ejemplo 8.71

```
\begin{displaymath}
  \sin^2 \theta + \cos^2 \theta = 1
\end{displaymath}
```

$$\sin^2\theta+\cos^2\theta=1$$

La expresión sin modificar las longitudes expuestas. Ejemplos cuando se modifican las longitudes:

Ejemplo 8.72

```
\thinmuskip = 0mu
\begin{displaymath}
  \sin^2 \theta + \cos^2 \theta = 1
\end{displaymath}
```

$$\sin^2\theta+\cos^2\theta=1$$

Ejemplo 8.73

```
\medmuskip = 0mu
\begin{displaymath}
  \sin^2 \theta + \cos^2 \theta = 1
\end{displaymath}
```

$$\sin^2\theta+\cos^2\theta=1$$

Ejemplo 8.74

```
\thickmuskip = 0mu
\begin{displaymath}
  \sin^2 \theta + \cos^2 \theta = 1
\end{displaymath}
```

$$\sin^2\theta+\cos^2\theta=1$$

Espacios veriticales

Antes y después de la expresión matemáticas

Para modificar los espacios antes y después de las expresiones matemáticas están definidos las longitudes:

Espacios verticales

```
\abovedisplayskip
\abovedisplayshortskip
\belowdisplayskip
\belowdisplayshortskip
```

Ejemplo 8.75

```
\documentclass{article}
\usepackage{amsmath}
\usepackage[paperwidth=10cm,paperheight=2.5cm,margin=1mm]{geometry}
\begin{document}
  \abovedisplayshortskip = 0pt
  \belowdisplayshortskip = 0pt
```

```
  \abovedisplayskip = 0pt
  \belowdisplayskip = 0pt
  Demostrar la identidad:
  \begin{equation}
    \sin^2 \theta + \cos^2 \theta = 1
  \end{equation}
  Demostrar la identidad:
  \begin{equation}
    \sin^2 \theta + \cos^2 \theta = 1
  \end{equation}
\end{document}
```

Demostrar la identidad:

$$\sin^2\theta + \cos^2\theta = 1 \quad (1)$$

Demostrar la identidad:

$$\sin^2\theta + \cos^2\theta = 1 \quad (2)$$

Espacio entre líneas

`\jot`

Para saltar a la siguiente línea, tanto en modo matemático y texto, se utiliza `\\[<lenght>]`. También podemos establecer la longitud `\jot` para cambiar la longitud entre líneas; por ejemplo,

Ejemplo 8.76

```
\begin{align*}
  x + y + z &= 1\\
  x &= 1 - y - z
\end{align*}
```

$$\begin{aligned} x+y+z&=1\\ x&=1-y-z \end{aligned}$$

Ejemplo 8.77

```
\jot=1cm
\begin{align*}
  x + y + z &= 1\\
  x &= 1 - y - z
\end{align*}
```

$$\begin{aligned} x+y+z&=1\\[1cm] x&=1-y-z \end{aligned}$$

Espacio entre líneas en array

Para establecer el espacio entre las líneas en un entorno `array` debemos recurrir al comando `\arraystretch`.

Ejemplo 8.78

```
\begin{displaymath}
  \begin{pmatrix}
    1 & 1 & 1 \\
    1 & 1 & 1 \\
    1 & 1 & 1
  \end{pmatrix}
\end{displaymath}
```

$$\begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}$$

Ejemplo 8.79

```
\renewcommand\arraystretch{1.5}
\begin{displaymath}
  \begin{pmatrix}
    1 & 1 & 1 \\
    1 & 1 & 1 \\
    1 & 1 & 1
  \end{pmatrix}
\end{displaymath}
```

$$\begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}$$

Para espaciados de una matriz también podemos utilizar al paquete `setspace`.

8.13 Estilos

Los estilos existentes de expresiones matemáticas son: `\displaystyle`,

Ejemplo 8.80

```
\begin{displaymath}
  \displaystyle
  f(\theta) = \int\left(\sin^2 \theta + \cos^2
  ↪  \theta\right)d\theta
\end{displaymath}
```

$$\displaystyle f(\theta) = \int\left(\sin^2 \theta + \cos^2 \theta\right)d\theta$$

; `\scriptstyle`,

Ejemplo 8.81

```
\begin{displaymath}
  \scriptstyle
  f(\theta) = \int\left(\sin^2 \theta + \cos^2
  ↪  \theta\right)d\theta
\end{displaymath}
```

$$\scriptstyle f(\theta) = \int\left(\sin^2 \theta + \cos^2 \theta\right)d\theta$$

; `\scriptscriptstyle`,

Ejemplo 8.82

```
\begin{displaymath}
  \scriptscriptstyle
  f(\theta) = \int\left(\sin^2 \theta + \cos^2
  ↪  \theta\right)d\theta
\end{displaymath}
```

$\scriptscriptstyle f(\theta) = \int\left(\sin^2 \theta + \cos^2 \theta\right)d\theta$

; \textstyle,

Ejemplo 8.83

```
\begin{displaymath}
  \textstyle
  f(\theta) = \int\left(\sin^2 \theta + \cos^2
  ↪  \theta\right)d\theta
\end{displaymath}
```

$\textstyle f(\theta) = \int\left(\sin^2 \theta + \cos^2 \theta\right)d\theta$

8.14 Puntos

Ejemplo 8.84

`$\cdots$`	$\cdots$

Ejemplo 8.85

`$\ldots$`	$\ldots$

Ejemplo 8.86

`$\vdots$`	$\vdots$

Ejemplo 8.87

`$\ddots$`	$\ddots$

Ejemplo 8.88

`\reflectbox{$\ddots$}`	⋰

Ejemplo 8.89

`$\dotsb$`	$\cdots$

Ejemplo 8.90

`$\dotsc$`	$\ldots$

Ejemplo 8.91

`$\dotsi$`	$\cdots$

Ejemplo 8.92

`$\dotsm$`	$\dotsm$

Ejemplo 8.93

`$\dotso$`	$\dotso$

8.15 Acentos

Acentos y sombreros más comunes

Ejemplo 8.94

`$\hat{\imath}$`	$\hat{\imath}$

Ejemplo 8.95

`$\acute{a}$`	$\acute{a}$

Ejemplo 8.96

`$\bar{a}$`	$\bar{a}$

Ejemplo 8.97

`$\vec{a}$`	$\vec{a}$

Llaves horizontales

Ejemplo 8.98

```
\begin{displaymath}
  E = \underbrace{\sqrt{2}\sqrt{2}\sqrt{2}\ldots}_{\text{multiplicado $n$ veces}}
\end{displaymath}
```

$$E = \underbrace{\sqrt{2}\sqrt{2}\sqrt{2}\ldots}_{\text{multiplicado n veces}}$$

Ejemplo 8.99

```
\begin{displaymath}
  E = \overbrace{\sqrt{2}\sqrt{2}\sqrt{2}\ldots}^{\text{multiplicado $n$ veces}}
\end{displaymath}
```

$$E = \overbrace{\sqrt{2}\sqrt{2}\sqrt{2}\ldots}^{\text{multiplicado n veces}}$$

Barras horizontales

Ejemplo 8.100

`$\bar{AB}$`	$\bar{AB}$

Ejemplo 8.101

`$\overline{AB}$`	$\overline{AB}$

Ejemplo 8.102

`$\underline{AB}$`	$\underline{AB}$

Vectores

Para escribir un vecto existe el comando `\vec`; también existe otro comando, `\vv`, definido en el paquete `esvect`.

Ejemplo 8.103

```
\begin{displaymath}
  \vec{a} = \vec{m} + \vec{n}
\end{displaymath}
```

$$\vec{a} = \vec{m} + \vec{n}$$

Ejemplo 8.104

```
\begin{displaymath}
  \vec{a} = \vec{m} + \vec{n}
\end{displaymath}
```

$$\vec{a} = \vec{m} + \vec{n}$$

8.16 Comandos underset y overset

Ejemplo 8.105

`$\underset{under}{baseline}$`	$\underset{under}{baseline}$

Ejemplo 8.106

`$\overset{over}{baseline}$`	$\overset{over}{baseline}$

8.17 Exponentes y subíndices

Para que exista una buena composición tipográfica sobre exponentes es necesario que realicemos cierta operación específica. Por ejemplo,

Ejemplo 8.107

```
\begin{displaymath}
  (((a^2)^2)^2)^2
\end{displaymath}
```

$$(((a^2)^2)^2)^2$$

Este fragmento de código imprime los exponentes incorrecto o no tan buenos, estéticamente. Para solucionar este problema debemos recurrir a las alturas dinámicas de los delimitadores.

Ejemplo 8.108

```
\begin{displaymath}
  \left(\left(\left(\left(a^{2}\right)^{2}\right)^{2}\right)^{2}
\end{displaymath}
```

$$\left(\left(\left(\left(a^{2}\right)^{2}\right)^{2}\right)^{2}\right)^{2}$$

Para obtener los subíndices debemos de utilizar el carácter `_`.

Ejemplo 8.109

```
$H_2O$
```

H_2O

Para indicar que un fragmento de código forma parte de un subíndice, debemos de utilizar lo que son los delimitadores de código.

Ejemplo 8.110

```
$C_{12}H_{22}O_{11}$
```

$C_{12}H_{22}O_{11}$

8.18 Operadores

Los operadores más conocidos en LaTeX se muestra en la tabla 8.14. Los operadores descritos en la tabla

Código	Símbolo	Código	Símbolo
`\int`	$\int$	`\intop`	$\intop$
`\smallint`	$\smallint$	`\oint`	$\oint$
`\prod`	$\prod$	`\sum`	$\sum$
`\coprod`	$\coprod$	`\bigcap`	$\bigcap$
`\bigcup`	$\bigcup$	`\bigsqcup`	$\bigsqcup$
`\bigwedge`	$\bigwedge$	`\bigvee`	$\bigvee$
`\bigoplus`	$\bigoplus$	`\bigotimes`	$\bigotimes$
`\bigodot`	$\bigodot$	`\biguplus`	$\biguplus$
`\bigcirc`	$\bigcirc$	`\bigtriangleup`	$\bigtriangleup$

Tabla 8.14

Operadores predefinidos de fontmath.ltx

tienen como límites encima y por debajo; por ejemplo,

Ejemplo 8.111

```
\begin{displaymath}
  \sum_{i=1}^{n} i^2
\end{displaymath}
```

$$\sum_{i=1}^{n} i^2$$

Los operadores que se detallaran ahora tienen como límite al subíndice y el superíndice; por ejemplo,

Ejemplo 8.112

```
\begin{displaymath}
  \sin^2\theta
\end{displaymath}
```

$$\sin^2\theta$$

Los operadores predefinidos en LaTeX se muestra en la tabla 8.15

Tabla 8.15 Operadores predefinidos de latex.ltx

Código	Símbolo	Código	Símbolo	Código	Símbolo
\log	log	\lim	lim	\sin	sin
\cos	cos	\tan	tan	\cot	cot
\csc	csc	\sup	sup	\ker	ker
\det	det	\gcd	gcd	\lg	lg
\limsup	limsup	\arcsin	arcsin	\arccos	arccos
\arctan	arctan	\coth	coth	\max	max
\inf	inf	\dim	dim	\exp	exp
\deg	deg	\ln	ln	\liminf	liminf
\sinh	sinh	\cosh	cosh	\tanh	tanh
\sec	sec	\min	min	\arg	arg
\hom	hom	\Pr	Pr	\bmod	mod

\DeclareMathOperator

Definir nuevos operadores se debe de recurir al \DeclareMathOperator; por ejemplo,

```
\DeclareMathOperator{\traz}{traz}
```

Los nuevos operadores declarados se deben de realizar en el preámbulo del documento Latex.

8.19 Letras griegas

\allowdisplaybreaks

\displaybreak

Las letras griegas se muestran en la tabla 8.16. Las letras griegas mayúsculas se muestran en la tabla 8.17.

8.20 Saltos de páginas

En las fórmulas matemáticas no se pueden tener saltos de página. Sin embargo existe el comando \allowdisplaybreaks para que admita saltos de páginas en fórmulas matemáticas. El comando \displaybreak se utiliza para realizar saltos de páginas en modo matemático.

Código	Símbolo	Código	Símbolo
\alpha	α	\beta	β
\gamma	γ	\delta	δ
\epsilon	ϵ	\zeta	ζ
\eta	η	\theta	θ
\iota	ι	\kappa	κ
\lambda	λ	\mu	μ
\nu	ν	\xi	ξ
\pi	π	\rho	ρ
\sigma	σ	\tau	τ
\upsilon	υ	\phi	ϕ
\chi	χ	\psi	ψ
\omega	ω	\varepsilon	ε
\vartheta	ϑ	\varkappa	$\varkappa$

Tabla 8.16

Código	Símbolo	Código	Símbolo
\Gamma	Γ	\Delta	Δ
\Theta	Θ	\Lambda	Λ
\Xi	Ξ	\Pi	Π
\Sigma	Σ	\Upsilon	Υ
\Phi	Φ	\Psi	Ψ
\Omega	Ω	\vartheta	ϑ

Tabla 8.17

Letras griegas mayúsculas

Ejemplo 8.113

```
\documentclass{article}
\usepackage{amsmath,cancel}
\allowdisplaybreaks
\usepackage{geometry}
\geometry{%
  paperwidth=14cm,left=1cm,right=1cm,
  paperheight=8cm,top=8mm,bottom=8mm,
}
\begin{document}
  \begin{align*}
    E & = \frac{a^{2} - 1}{n^{2} + an}\left( \frac{1}{1-\dfrac{1}{n}} -
    ↪  1\right)\frac{a-an^{3}-n^{4}+n}{1-a^{2}}  \\
    E & = \frac{-\cancel{\left(1-a^{2}\right)}}{n\left(n + a \right)}
    ↪  \left(\dfrac{1}{\dfrac{n-1}{n}}-1\right) \frac{a + n - n^{3}\left(a +
    ↪  n\right)}{\cancel{1-a^{2}}}       \\
    E & = \frac{-1}{\cancel{n}\cancel{\left(n + a\right)}} \left( \frac{1-\dfrac{n-1}{n}}{\dfrac{n
    ↪  -1}{\cancel{n}}}\right) \cancel{\left(a + n\right)} \left( 1 - n^{3}\right) \\
    E & = -1\left(\frac{1}{n\left(n-1\right)}\right) \left(1-n\right)\left(1+n+n^2\right) \\
    E & = \left(\frac{1}{n\cancel{\left(1-n\right)}}\right)
    ↪  \cancel{\left(1-n\right)}\left(1+n+n^2\right) \\
    E & = \frac{1 + n + n^2}{n}
```

```
  \end{align*}
\end{document}
```

$$
\begin{aligned}
E &= \frac{a^2-1}{n^2+an}\left(\frac{1}{1-\frac{1}{n}}-1\right)\frac{a-an^3-n^4+n}{1-a^2}\\
E &= \frac{-\cancel{(1-a^2)}}{n\,(n+a)}\left(\frac{1}{\frac{n-1}{n}}-1\right)\frac{a+n-n^3\,(a+n)}{\cancel{1-a^2}}\\
E &= \frac{-1}{\cancel{n}\cancel{(n+a)}}\left(\frac{1-\frac{n-1}{n}}{\frac{n-1}{\cancel{n}}}\right)\cancel{(a+n)}\left(1-n^3\right)\\
E &= -1\left(\frac{1}{n\,(n-1)}\right)(1-n)\left(1+n+n^2\right)
\end{aligned}
$$

$$
\begin{aligned}
E &= \left(\frac{1}{n\cancel{(1-n)}}\right)\cancel{(1-n)}\left(1+n+n^2\right)\\
E &= \frac{1+n+n^2}{n}
\end{aligned}
$$

8.21 Entornos de alineación del paquete amsmath

Casi todos los entornos de alineado siguien la sintaxis:

```
\begin{<name environment>}
  <expression> &= <expression>    &    <expression> &= <expression>\\
  <expression> &= <expression>    &    <expression> &= <expression>
\end{<name environment>}
```

align

Entorno align

La sintaxis del entorno `align` es:

```
\begin{align}
  <expression> &= <expression>    &    <expression> &= <expression>\\
  <expression> &= <expression>    &    <expression> &= <expression>
\end{align}
```

Ejemplo 8.114

```
\begin{align}
  x^2 + y^2 &= 1\\
  x &= \sqrt{1-y^2}
\end{align}
```

$$x^2+y^2=1 \tag{8}$$

$$x=\sqrt{1-y^2} \tag{9}$$

Ejemplo 8.115

```
\begin{align*}
  x^2 + y^2 &= 1\\
  x &= \sqrt{1-y^2}
\end{align*}
```

$$\begin{aligned} x^2+y^2 &= 1\\ x &= \sqrt{1-y^2} \end{aligned}$$

Ejemplo 8.116

```
\begin{align*}
  x^2 + y^2 &= 1            &       x &= a+b+c\\
  x         &= \sqrt{1-y^2} &         y &= a-b-c
\end{align*}
```

$$\begin{aligned} x^2+y^2 &= 1 & x &= a+b+c\\ x &= \sqrt{1-y^2} & y &= a-b-c \end{aligned}$$

Comando intertext

`\intertext`

El comando `\intertext` permite insertar texto en entornos de alineado como `<align>`.

Ejemplo 8.117

```
\begin{align}
  x^2 + y^2 &= 1\\
  \intertext{Despejando $x$}.
  x &= \sqrt{1-y^2}
\end{align}
```

$$x^2+y^2 = 1 \tag{10}$$

Despejando x

$$.x = \sqrt{1-y^2} \tag{11}$$

Entorno alignat

La sintaxis del entorno `alignat`:

alignat

```
\begin{alignat}{<number of columns>}
  <expression> &= <expression>    &     <expression> &= <expression>\\
  <expression> &= <expression>    &     <expression> &= <expression>
\end{alignat}
```

Ejemplo 8.118

```
\begin{alignat}{3}
  x_{12}&=2 & x_{13}&=3 & x_{14}&=4\\
  x_{23}&=5 & x_{24}&=6 & x_{34}&=7\\
```

```
  x_{45}&=8 & x_{56}&=9 & x_{67}&=10
\end{alignat}
```

$$x_{12} = 2x_{13} = 3x_{14} = 4 \quad (12)$$

$$x_{23} = 5x_{24} = 6x_{34} = 7 \quad (13)$$

$$x_{45} = 8x_{56} = 9x_{67} = 10 \quad (14)$$

Entorno flalign

xalignat y xxalignat

Este entorno ha reemplazado a los entornos `xalignat` y `xxalignat`. La sintaxis de este entorno es:

```
\begin{flalign}
  <expression> &= <expression>    &    <expression> &= <expression>\\
  <expression> &= <expression>    &    <expression> &= <expression>
\end{flalign}
```

Ejemplo 8.119

```
\begin{flalign}
  x_{12}&=2 & x_{13}&=3 & x_{14}&=4\\
  x_{23}&=5 & x_{24}&=6 & x_{34}&=7\\
  x_{45}&=8 & x_{56}&=9 & x_{67}&=10
\end{flalign}
```

$$x_{12} = 2 \qquad x_{13} = 3 \qquad x_{14} = 4 \quad (15)$$

$$x_{23} = 5 \qquad x_{24} = 6 \qquad x_{34} = 7 \quad (16)$$

$$x_{45} = 8 \qquad x_{56} = 9 \qquad x_{67} = 10 \quad (17)$$

Ejemplo 8.120

```
\begin{flalign}
  f(x) &= \int\frac{1}{x^2}dx
\end{flalign}
```

$$f(x) = \int \frac{1}{x^2} dx \quad (18)$$

Ejemplo 8.121

```
\begin{flalign*}
  f(x) &= \int\frac{1}{x^2}dx
\end{flalign*}
```

$$f(x) = \int \frac{1}{x^2} dx$$

Entorno aligned

aligned

Es similar al entorno `array` y tiene que ser parte de otro entorno matemático.

```
\begin{aligned}
  <expression> &= <expression>    &     <expression> &= <expression>\\
  <expression> &= <expression>    &     <expression> &= <expression>
\end{aligned}
```

Ejemplo 8.122

```
\begin{displaymath}
  \begin{aligned}
    x_{12}&=2 & x_{13}&=3 & x_{14}&=4\\
    x_{23}&=5 & x_{24}&=6 & x_{34}&=7\\
    x_{45}&=8 & x_{56}&=9 & x_{67}&=10
  \end{aligned}
\end{displaymath}
```

$$\begin{aligned} x_{12}&=2 & x_{13}&=3 & x_{14}&=4\\ x_{23}&=5 & x_{24}&=6 & x_{34}&=7\\ x_{45}&=8 & x_{56}&=9 & x_{67}&=10 \end{aligned}$$

8.22 Otros entornos de amsmath

gather

Entorno gather

Es un entorno de varias líneas de expresiones matemáticas y están centreados.

Ejemplo 8.123

```
\begin{gather}
  x^2 + y^2 = 1\\
  x = \sqrt{1-y^2}
\end{gather}
```

$$x^2+y^2=1 \qquad (19)$$

$$x=\sqrt{1-y^2} \qquad (20)$$

La versión con estrella de este entorno no genera la numeración de ecuaciones.

gathered

Entorno gathered

Este entorno es similar a los entornos `aligned` y `alignat` solo que este entorno ocupa todo el espacio horizontal y necesariamento tiene que estar dentro de un entorno matemático.

Ejemplo 8.124

```
\begin{align*}
  \begin{gathered}
    x^2 + y^2 = 1\\
    x = \sqrt{1-y^2}
  \end{gathered}
\end{align*}
```

$$\begin{gathered} x^2 + y^2 = 1\\ x = \sqrt{1-y^2} \end{gathered}$$

Este entorno, por default, centrea verticalmente las ecuaciones matemáticas, pero nosotros podemos indicar que esté alineado por parte de arriba (t) o por debajo (b).

Ejemplo 8.125

```
\begin{align*}
  \rule{2cm}{0.5pt}
  \begin{gathered}[t]
    x^2 + y^2 = 1\\
    x = \sqrt{1-y^2}
  \end{gathered}
  \hspace{1cm}
  \begin{gathered}[c]
    x^2 + y^2 = 1\\
    x = \sqrt{1-y^2}
  \end{gathered}
  \hspace{1cm}
  \begin{gathered}[b]
    x^2 + y^2 = 1\\
    x = \sqrt{1-y^2}
  \end{gathered}
  \rule{2cm}{0.5pt}
\end{align*}
```

$$\rule{2cm}{0.5pt} \begin{gathered}[t] x^2 + y^2 = 1\\ x = \sqrt{1-y^2} \end{gathered} \hspace{1cm} \begin{gathered}[c] x^2 + y^2 = 1\\ x = \sqrt{1-y^2} \end{gathered} \hspace{1cm} \begin{gathered}[b] x^2 + y^2 = 1\\ x = \sqrt{1-y^2} \end{gathered} \rule{2cm}{0.5pt}$$

multline

Entorno multline

Es un entorno de varias líneas, en donde la primera línea se alinea a la izquierda y la segunda, la tercera, la cuarta y hasta la penúltima línea están centreados, y finalmente, la última línea se alinea hacia la derecha.

Ejemplo 8.126

```
\begin{multline*}
  1-\frac{x}{1 !}+\frac{x(x-1)}{2 !}
  ↪  -\frac{x(x-1)(x-2)}{3 !}=\\
  = \frac{(x-1)(x-2)}{2} -
  ↪  \frac{x(x-1)(x-2)}{6} =\\
  = -\frac{(x-1)(x-2)(x-3)}{3!}
\end{multline*}
```

$$1-\frac{x}{1!}+\frac{x(x-1)}{2!}-\frac{x(x-1)(x-2)}{3!}= \\ =\frac{(x-1)(x-2)}{2}-\frac{x(x-1)(x-2)}{6}= \\ =-\frac{(x-1)(x-2)(x-3)}{3!}$$

Existe la versión con estrella y sin estrella de este entorno.

split

Entorno split

Este entorno puede ser utilizado solamente dentro de un entorno matemático. Cuando no se utiliza el signo ampersand dentro del entorno `split` se alinea las ecuaciones o las expresiones matemáticas a la izquierda y, mientras tanto, cuando se utiliza el signo ampersand la alineación sucede hacia la izquierda.

Ejemplo 8.127

```
\begin{align*}
  \begin{split}
    \left|\int_{0}^{1}(f(x)-g(x)) \mathrm{d} x\right| + \left|\int_{1}^{2}(g(x)-h(x)) \mathrm{d}
    ↪  x\right|\\
    \left|\int_{0}^{1}\left(x^{2}-3 x\right) \mathrm{d} x\right| + \left|\int_{1}^{2}\left(x^{2}-5
    ↪  x+6\right) \mathrm{d} x\right|\\
    \left|\frac{x^{3}}{3} - \frac{3}{2} x^{2}\right|_{0}^{1} + \left|\frac{x^{3}}{3} - \frac{5}{2}
    ↪  x^{2}+6 x\right|_{1}^{2}\\
    \left|\frac{1}{3} - \frac{3}{2}\right| + \left|\frac{8}{3} - \frac{20}{2} + 12 -
    ↪  \left(\frac{1}{3}-\frac{5}{2}+6\right)\right|\\
    \left|-\frac{7}{6}\right| + \left|\frac{14}{3}-\frac{23}{6}\right|=\frac{7}{6}+\frac{5}{6}=2
  \end{split}
\end{align*}
```

$$
\begin{aligned}
&\left|\int_{0}^{1}(f(x)-g(x)) \mathrm{d} x\right| + \left|\int_{1}^{2}(g(x)-h(x)) \mathrm{d} x\right|\\
&\left|\int_{0}^{1}\left(x^{2}-3 x\right) \mathrm{d} x\right|+\left|\int_{1}^{2}\left(x^{2}-5 x+6\right) \mathrm{d} x\right|\\
&\left|\frac{x^{3}}{3} - \frac{3}{2} x^{2}\right|_{0}^{1} + \left|\frac{x^{3}}{3} - \frac{5}{2} x^{2}+6 x\right|_{1}^{2}\\
&\left|\frac{1}{3} - \frac{3}{2}\right| + \left|\frac{8}{3} - \frac{20}{2}+12 - \left(\frac{1}{3} - \frac{5}{2}+6\right)\right|\\
&\left|-\frac{7}{6}\right| + \left|\frac{14}{3} - \frac{23}{6}\right|=\frac{7}{6} + \frac{5}{6}=2
\end{aligned}
$$

Ejemplo 8.128

```
\begin{align*}
  \begin{split}
    E &= \left|\int_{0}^{1}(f(x)-g(x)) \mathrm{d} x\right| + \left|\int_{1}^{2}(g(x)-h(x)) \mathrm{d}
    ↪  x\right|\\
    &=\left|\int_{0}^{1}\left(x^{2}-3 x\right) \mathrm{d} x\right|+\left|\int_{1}^{2}\left(x^{2}-5
    ↪  x+6\right) \mathrm{d} x\right|\\
    &=\left|\frac{x^{3}}{3} - \frac{3}{2} x^{2}\right|_{0}^{1} + \left|\frac{x^{3}}{3} - \frac{5}{2}
    ↪  x^{2}+6 x\right|_{1}^{2}\\
    &=\left|\frac{1}{3} - \frac{3}{2}\right| + \left|\frac{8}{3} - \frac{20}{2}+12 - \left(\frac{1}{3}
    ↪  - \frac{5}{2}+6\right)\right|\\
    &=\left|-\frac{7}{6}\right| + \left|\frac{14}{3} - \frac{23}{6}\right|=\frac{7}{6} + \frac{5}{6}=2
  \end{split}
\end{align*}
```

$$
\begin{split}
E &= \left|\int_{0}^{1}(f(x)-g(x)) \mathrm{d} x\right| + \left|\int_{1}^{2}(g(x)-h(x)) \mathrm{d} x\right|\\
&=\left|\int_{0}^{1}\left(x^{2}-3 x\right) \mathrm{d} x\right|+\left|\int_{1}^{2}\left(x^{2}-5 x+6\right) \mathrm{d} x\right|\\
&=\left|\frac{x^{3}}{3} - \frac{3}{2} x^{2}\right|_{0}^{1} + \left|\frac{x^{3}}{3} - \frac{5}{2} x^{2}+6 x\right|_{1}^{2}\\
&=\left|\frac{1}{3} - \frac{3}{2}\right| + \left|\frac{8}{3} - \frac{20}{2}+12 - \left(\frac{1}{3} - \frac{5}{2}+6\right)\right|\\
&=\left|-\frac{7}{6}\right| + \left|\frac{14}{3} - \frac{23}{6}\right|=\frac{7}{6} + \frac{5}{6}=2
\end{split}
$$

cases

Entorno cases

Ejemplo 8.129

```
\begin{align*}
  |x| &= \begin{cases}
    x & \text{si } x \geq 0 \\
    -x & \text{si } x < 0 \\
  \end{cases}
\end{align*}
```

$$|x| = \begin{cases} x & \text{si } x \geqslant 0 \\ -x & \text{si } x < 0 \end{cases}$$

8.23 Raices con amsmath

La sintaxis es para las raices es:

`\sqrt`

```
\sqrt[%
  \leftroot{<number>}
  \uproot{<number>}
  \rightroot{<number>}
  \downroot{<number>}
  <root>
]{<expression>}
```

Ejemplo 8.130

```
\begin{displaymath}
  \sqrt[\sqrt{a}]{a}
\end{displaymath}
```

$$\sqrt[\sqrt{a}]{a}$$

Ejemplo 8.131

```
\begin{displaymath}
  \sqrt[\uproot{2}\sqrt{a}]{a}
\end{displaymath}
```

$$\sqrt[\uproot{2}\sqrt{a}]{a}$$

8.24 Límites

Los límites en displaymath se realizan de manera similar que modo linea.

Ejemplo 8.132

```
\begin{displaymath}
  \sum\limits_{i=1}^{n}i^{2}
\end{displaymath}
```

$$\sum\limits_{i=1}^{n}i^{2}$$

Límites múltiples

Para realizar lĩimites múltiples existen los comandos `\atop` y `\substack`.

Ejemplo 8.133

```
\begin{align*}
  \sum\limits_{%
    \substack{%
      1 \leq i \leq p\\
      1 \leq j \leq q\\
      1 \leq k \leq r
    }%
  } a_{ij}b_{ik}c_{ki}
\end{align*}
```

$$\sum\limits_{\substack{1\leqslant i\leqslant p\\1\leqslant j\leqslant q\\1\leqslant k\leqslant r}} a_{ij}b_{ik}c_{ki}$$

8.25 Flechas

Flechas no extensibles

Las flechas no extensibles se muestran en la tabla 8.18.

`\rightarrow`	$\rightarrow$	`\Rightarrow`	$\Rightarrow$
`\leftarrow`	$\leftarrow$	`\Leftarrow`	$\Leftarrow$
`\uparrow`	$\uparrow$	`\Uparrow`	$\Uparrow$
`\downarrow`	$\downarrow$	`\Downarrow`	$\Downarrow$
`\updownarrow`	$\updownarrow$	`\Updownarrow`	$\Updownarrow$
`\nwarrow`	$\nwarrow$	`\nearrow`	$\nearrow$
`\swarrow`	$\swarrow$	`\searrow`	$\searrow$
`\nleftarrow`	$\nleftarrow$	`\nLeftarrow`	$\nLeftarrow$
`\nrightarrow`	$\nrightarrow$	`\nRighttarrow`	$\nRightarrow$
`\leftrightarrow`	$\leftrightarrow$	`\Leftrightarrow`	$\Leftrightarrow$
`\longleftarrow`	$\longleftarrow$	`\Longleftarrow`	$\Longleftarrow$
`\longrightarrow`	$\longrightarrow$	`\Longrightarrow`	$\Longrightarrow$
`\longleftrightarrow`	$\longleftrightarrow$	`\Longleftrightarrow`	$\Longleftrightarrow$
`\longmapsto`	$\longmapsto$		

Tabla 8.18

Flechas no extensibles

`\xrightarrow`

`\xleftarrow`

Flechas extensibles

Las flechas extensibles se obtienen con el comando `\xrightarrow` y `\xleftarrow`. Sintaxis del comando `\xrightarrow`:

```
\xrightarrow[<text below>]{<text above>}
```

Ejemplo 8.134

```
\begin{displaymath}
  \xrightarrow[\text{texto abajo}]{\text{texto arriba}}
\end{displaymath}
```

$$\xrightarrow[\text{texto abajo}]{\text{texto arriba}}$$

Ejemplo 8.135

```
\begin{displaymath}
  \xleftarrow[\text{texto abajo}]{\text{texto arriba}}
\end{displaymath}
```

$$\xleftarrow[\text{texto abajo}]{\text{texto arriba}}$$

8.26 Otros paquetes matemáticos

accents	alphalph	amsart	amsbook
amsbsy	amscdx	amscls	amsfonts
amsLaTeX	amsltx11	amsmath	amsppt
amsppt1	amsproc	amssym (plain TeX)	amssymb (LaTeX)
amstex (Plain TeX)	amstext	amsthm	bez123
bitfield	brclc	breqn	cancel
cases	comma	datenumber	diagxy
doublestroke	easyeqn	easybmat	easymat
eqnarray	esvect	fixmath	ftlpoint
icomma	leftidx	mathdots	mathtools
mathematica	mil3	mtbe	Nath
numprint	random	romannum	TeXaide

Paquete cancel

\cancel

Este paquete es utilizado para realizar las simplificaciones o las cancelaciones en expresiones matemáticas; por ejemplo,

Ejemplo 8.136

```
\begin{displaymath}
  f(x)=\frac{\left(x^{2}+1\right)\cancel{(x-1)}}{\cancel{(x-1)}(x+1)}
\end{displaymath}
```

$$f(x)=\frac{\left(x^{2}+1\right)\cancel{(x-1)}}{\cancel{(x-1)}(x+1)}$$

Para utilizar el comando \cancel es necesario importar o utilizar el paquete cancel (\usepackage {cancel}).

Para mayor información acerca de este paquete, se recomienda revisar la documentación oficial de este paquete.

Paquete empheq

empheq

Este paquete es utilizado para poner las expresiones matemáticas dentro de cajas a colores. Por ejemplo,

Ejemplo 8.137

```
\documentclass{article}
\usepackage{amsmath,empheq,tcolorbox}
\usepackage{geometry}
\geometry{%
  paperwidth=10cm,left=1cm,right=1cm,
  paperheight=5cm,top=8mm,bottom=8mm,
}
\begin{document}
  \begin{empheq}[box={\fboxsep=10pt\colorbox{yellow}}]{align}
    f(x)=\int_1^{\infty}\frac{1}{x^2}\,\mathrm{d}x=1
  \end{empheq}
\end{document}
```

$$f(x)=\int_1^{\infty}\frac{1}{x^2}\,\mathrm{d}x=1 \qquad (1)$$

El paquete empheq podemos combinar con el paquete tcolorbox.

GRÁFICAS EN LATEX

En este capítulo aprenderemos a importar gráficas en LATEX y a generarlas con paquetes como tikz (también existen los paquetes pgfplots y chemfig). Este último paquete se utiliza para realizar las representaciones de compuestos químicos, como el benceno.

9.1 Paquete graphicx

Sintaxis para utilizar el paquete graphicx:

```
\usepackage{graphicx}
```

Para insertar las gráficas en un documento LATEX debemos recurrir al paquete graphicx. Este paquete proporciona el comando,

```
\includegraphics[<options>]{<ruta de la gráfica o imagen>}
```

que nos permite importar gráficas desde una carpeta ubicado en el mismo directorio que el documento LATEX.

\includegraphics

Opciones del comando \includegraphics:

1. scale=<number>: Hace un escalado de la gráfica.
2. width=<length>: Establece el ancho de la gráfica.
3. height=<length>: Establece la altura de la gráfica.
4. totalheight=<length>: Establece la altura de la gráfica más la profundidad de la gráfica.
5. keepaspectratio=<true/false>: Mantiene la relación de aspecto de la gráfica.
6. angle=<number>: Establece el ángulo de rotación de la gráfica.
7. origin=<location>: Establece el punto de rotación de la gráfica.
8. draft=<true/false>: Evita la importación de la gráfica, pero crea una caja con las dimensiones de la gráfica (esta opción se utiliza para acelerar el procesamiento).
9. clip=<true/false>: Excluye lo que esté fuera del cuadro delimitador.

10 `bb=<llx lly urx ury>`: Introduce las coordenadas del cuadro delimitador, que son dado por defecto en puntos (1/72 pulgada).

11 `viewport=<llx lly urx ury>`: Establece el cuadro delimitador respecto al borde inferior izquierdo del cuadro delimitador existente; se utiliza con clip para seleccionar una parte de la imagen (o para limpiar los bordes innecesarios).

12 `trim=<dllx dlly durx dury>`: Reduce el cuadro delimitador en la cantidad especificada.

Cuando tenemos las gráficas en una ruta determinada, por ejemplo, en `images/`, para no estar repitiendo el comando, como:

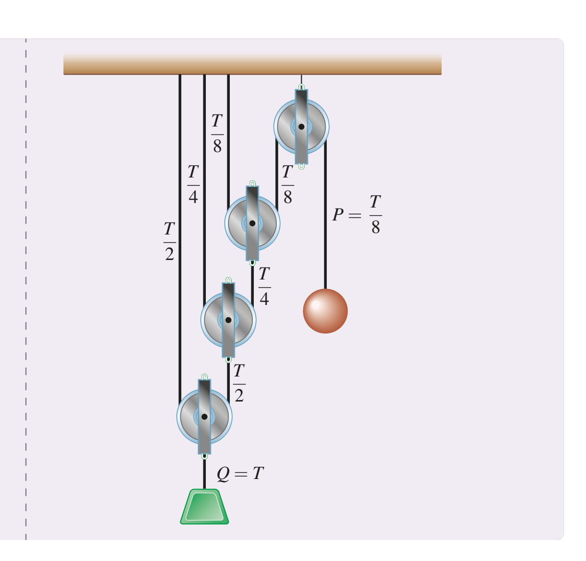
```
\includegraphics{images/grafica1}
\includegraphics{images/grafica2}
\includegraphics{images/grafica3}
```

Existe el comando `\graphicspath` (en este caso, `\graphicspath{images/}`) que nos permite cambiar la ruta de la gráfica (este comando podemos invocar en el preámbulo del documento LaTeX). Realizado esta operación ahora simplemente podemos realizar las siguientes tareas.

```
\graphicspath{images/}
\includegraphics{grafica1}
\includegraphics{grafica2}
\includegraphics{grafica3}
```

Ejemplo 9.1

```
\includegraphics[width=5cm]{images/grafica1}
```



Los formatos de imágenes recomendados que se puede utilizar en LaTeX son el jpg, png y pdf, aunque también podemos utilizar gráficas en otros formatos como eps.

9.2 Subfiguras

Para crear subfiguras es recomendable utilizar el paquete `subfigure`, para ello es necesario importar este paquete.

Ejemplo 9.2

```
\documentclass{article}
\usepackage[paperwidth=11cm,paperheight=7cm, margin=1mm]{geometry}
\usepackage{subfigure,graphicx}
\begin{document}
  \begin{figure}
    \centering
    \subfigure[Subfigura 1]{%
      \includegraphics[width=5cm]{images/grafica1}
      \label{fig:subfigura1}%
    }
    \subfigure[Subfigura 2]{%
      \includegraphics[width=5cm]{images/grafica1}
      \label{fig:subfigura2}%
    }
  \end{figure}
  Referencias: \ref{fig:subfigura1} y \ref{fig:subfigura2}
\end{document}
```

Referencias: 1(a) y 1(b)

(a) Subfigura 1

(b) Subfigura 2

9.3 Generación de gráficas

Las gráficas para utilizar en LaTeX podemos generarlas con diferentes programas como Inkscape, Adobe Illustrator y otros que tienen la capacidad de exportar gráficas o figuras en formatos como png, jpg, entre otros. Se recomienda exportar las gráficas desde programas como Inkscape en formatos vectoriales como PDF, ya que la calidad de la imagen no se pierde al utilizarla en LaTeX.

Inkscape

Este programa es un editor de gráficos vectoriales de código abierto (Open Source), similar a Adobe Illustrator. Las tipografías de LaTeX se pueden incluir mediante la extensión TexText.

Adobe Illustrator

Este programa, a diferencia de Inkscape, es de pago, pero es un programa profesional que permite crear gráficas vectoriales. Para incluir las composiciones tipográficas de LaTeX se debe construir un script en Javascript (también se pueden incluir gráficas generadas por tikz).

```
var pdflatexexe="pdflatex.exe";  // Add full path if necessary

// determining the local temporary directory
var temppath=Folder.temp.fsName;  // path already in Windows syntax: c:\...
var i=temppath.indexOf("Temporary Internet Files");
if(i>=0) temppath=temppath.substr(0,i+4);
//temppath should now contain something like C:\Documents and
↪  Settings\<user>\Local Settings\Temp

// remember the last user input in a text file
var lastcode=""
var lastcodefile=File(temppath+"\\latex2illustrator_lastcode.txt");
if(lastcodefile.exists)
  {
  lastcodefile.open("r");
  lastcode=lastcodefile.read();
  lastcodefile.close();
  }

// prompt for user input
var latexcode=prompt("Introducir codigo LaTex",lastcode,"LaTeX");
if(latexcode!=null)
  {
  lastcodefile.open("w");
  lastcodefile.write(latexcode);
  lastcodefile.close();

  // add latex header etc. to create a complete latex document
  var latexfile=new File(temppath+'\\latex2illustrator.tex');
  latexfile.open("w");
  latexfile.writeln("\\documentclass{standalone}");
```

```
// add or remove additional latex packages here
latexfile.writeln("\\usepackage{amsmath,amsfonts,amssymb,amsthm,latexsym,c
↪ ancel}");
latexfile.writeln("\\usepackage{pdfpages}");
//latexfile.writeln("\\usepackage[d]{esvect}");
latexfile.writeln("\\usepackage{lmodern}");
latexfile.writeln("\\usepackage{pslatex}");
//latexfile.writeln("\\usepackage{pifont}"); //ding{192=1, 193=2...}
latexfile.writeln("\\usepackage[latin1]{inputenc}");
latexfile.writeln("\\usepackage[T1]{fontenc}");
latexfile.writeln("\\usepackage{tgtermes}");
latexfile.writeln("\\usepackage{tikz}");
latexfile.writeln("\\usetikzlibrary{arrows.meta}");
latexfile.writeln("\\newcommand{\\grados}{^{\\circ}}");
latexfile.writeln("\\newcommand{\\unit}[1]{\\mathrm{#1}}");
latexfile.writeln("\\begin{document}");
latexfile.writeln("\\pagestyle{empty}"); // no page number
latexfile.writeln("\\small");
//commands
latexfile.writeln("\\newcommand\\vv[1]{");
latexfile.writeln("    \\begin{tikzpicture}[baseline=(arg.base)]");
latexfile.writeln("        \\node[inner xsep=0pt, inner ysep=1.5pt] (arg)
↪ {$#1$};");
latexfile.writeln("        \\draw[-{Stealth[length=2.15pt, width=2pt]}]
↪ ([yshift=-0.2pt,xshift=0.75pt]arg.north west) --
↪ ([yshift=-0.2pt,xshift=0pt]arg.north east) --++(0.5pt,0);");
latexfile.writeln("    \\end{tikzpicture}%");
latexfile.writeln("}");
//
latexfile.writeln(latexcode);
latexfile.writeln("\\end{document}");
latexfile.close();

var pdffile=File(temppath+"\\latex2illustrator.pdf");
if(pdffile.exists)
   pdffile.remove();

// create a batch file calling latex
var batchfile=new File(temppath+'\\latex2illustrator.bat');
batchfile.open("w");
```

```
batchfile.writeln(pdflatexexe+' -aux-directory="'+temppath+'"
↪ -include-directory="'+temppath+'" -output-directory="'+temppath+'"
↪ "'+temppath+'\\latex2illustrator.tex"');
//batchfile.writeln('pause');
batchfile.writeln('del "'+temppath+'\\latex2illustrator.bat"');
batchfile.close();
batchfile.execute();

//var batfile=File(temppath+"\\latex2illustrator.bat");
i//f(batfile.exists)
//  batfile.remove ();

for(; batchfile.exists; )
// wait until the batch file has removed itself

var pdffile=File(temppath+"\\latex2illustrator.pdf");
if(pdffile.exists)
  {
  // import pdf file into the current document
  var
↪ grp=app.activeDocument.activeLayer.groupItems.createFromFile(pdffile);
  // The imported objects are grouped twice. Now move the subgroup
  // items to the main group and skip the last item which is the page fram
e

  for( var i=grp.pageItems[0].pageItems.length; --i>=0; )
   grp.pageItems[0].pageItems[i].move(grp,ElementPlacement.PLACEATEND);

  var last = grp.pageItems.length - 1;
  if (last >= 0 && grp.pageItems[last].typename == 'PathItem')
      grp.pageItems[last].remove();

  // Move the imported objects to the center of the current view.
  grp.translate(app.activeDocument.activeView.centerPoint[0]-grp.left,
↪ app.activeDocument.activeView.centerPoint[1]-grp.top);
 ++)
//      objs[i].selected=true;
//var docSelected=tempDoc.selection;
//for(var i=0; i<docSelected.length; i++)
//     {
//     docSelected[i].selected=false;
//
↪ newItem=docSelected[i].duplicate(targetDoc,ElementPlacement.PLACEATEND);
```

```
//    }
//tempDoc.saved=true;
//tempDoc.close();
   }
  else
    alert("File "+temppath+"\\"+pdffile.name+" could not be created. LaTeX
↪  error?");
  }

//grp.pageItems.removeAll();

//var targetDoc=app.activeDocument;
//var tempDoc=open(File(temppath));
//var objs=tempDoc.activeLayer.pageItems;
//for(var i=0; i<objs.length; i
```

Para graficar las dimensiones (cotas) de longitudes en el programa Adobe illustrator se recomienda utilizar el plugin `cadtools`.

9.4 Iniciación con tikz

El paquete tikz de LaTeX es una herramienta de creación de gráficos vectoriales que permite a los usuarios crear gráficos de alta calidad y alta resolución. El paquete tikz está basado en el concepto de dibujo por medio de código, lo que significa que los usuarios pueden utilizar comandos de código para crear gráficos y figuras en su documento LaTeX. Esto permite a los usuarios tener un control preciso sobre la apariencia y formato de sus gráficos y figuras, y permite la creación de gráficos y figuras complejas de forma rápida y sencilla. Además, el paquete tikz se integra de manera fluida con el resto del documento LaTeX, lo que permite a los usuarios incluir gráficos y figuras en su documento de manera sencilla y sin problemas. En resumen, la filosofía del paquete tikz de LaTeX es proporcionar una herramienta poderosa y fácil de usar para la creación de gráficos y figuras de alta calidad en documentos LaTeX.

Sintaxis para usar paquete TikZ:

```
\usepackage{tikz}
```

El paquete TikZ proporciona sólo las herramientas básicas para realizar gráficos, pero existen otras herramientas adicionales que facilitan la realización de gráficos. Para acceder a estas herramientas es necesario importar o utilizar librerías TikZ. Sintaxis para importar o utilizar una librería TikZ:

```
\usetikzlibrary{<nombre librería>}
```

Para realizar gráficas en tikz podemos invocar el entorno `tikzpicture` o simplemente podemos utilizar el comando `\tikz`:

```
\begin{tikzpicture}[<opciones>]
  <código tikz>
\end{tikzpicture}
\tikz[<opciones>]<código tikz>
\tikz{<código tikz>}
```

Pgfkeys

pgfkeys es una herramienta de configuración y estilo incluida en el paquete TikZ para el lenguaje de formateo TeX. Permite a los usuarios definir y modificar opciones de configuración y estilo en sus gráficos TikZ.

Para utilizar pgfkeys, primero debes incluir el paquete TikZ en tu documento TeX.

Luego, puedes utilizar pgfkeys para definir y modificar opciones de configuración y estilo en tus gráficos TikZ de la siguiente manera:

```
\pgfkeys{/tikz/opción1/.initial=valor1, opción2/.initial=valor2, ...}
```

Esto define las opciones "opción1", "opción2", etc. con sus valores iniciales "valor1", "valor2", etc. Luego, puedes modificar el valor de estas opciones en cualquier parte de tu código TikZ con la siguiente sintaxis:

```
\tikzset{opción1=nuevo valor1, opción2=nuevo valor2, ...}
```

Esto cambiará el valor de las opciones "opción1", "opción2", etc. a "nuevo valor1", "nuevo valor2", etc. para el gráfico actual.

pgfkeys es una herramienta muy útil para la personalización de gráficos TikZ y permite un control preciso sobre la apariencia y el comportamiento de los gráficos. Si necesitas más información o ejemplos de cómo utilizar pgfkeys, no dudes en hacer otra pregunta.

Tikz

Especificación de puntos

En tikz existen ciertas maneras para especificar coordenadas de un punto.

para trazar algún línea o gráfica, se debe de utilizar el comando `\draw`.

Al igual que en geometría analítica, un punto se puede especificar par ordenado en el sistema cartesiano de dos dimensiones; por ejemplo, $(1cm, 2cm) \equiv (1,2)$ las pares ordenadas indican los puntos del sistema cartesiano, es decir, el punto se ubica a $1\,\mathrm{cm}$ del origen de coordenadas en el eje x y a $2\,cm$ del origen de coordenadas en el eje y. De la misma manera, se ubican los puntos de la forma $(1,2,5)$ en el espacio euclideano.

Ejemplo 9.3



```
\begin{center}
  \begin{tikzpicture}
    \draw (0,0) grid (2,2);
    \draw (1.5,1.5) circle (2pt);
  \end{tikzpicture}
\end{center}
```

El par `(30:2cm)` determina la coordenada polar de un punto, es decir, el punto se ubica a 2 cm del origen a un ángulo de 30 grados (medidos en sentido centrado de las manecillas de reloj) desde el eje x positivo.

Ejemplo 9.4

```
\begin{center}
  \begin{tikzpicture}
    \draw (0,0) grid (2,2);
    \draw (30:2cm) circle (2pt);
  \end{tikzpicture}
\end{center}
```

Las coordenadas relativas se las específica de la siguiente forma `++(3,5)`; por ejemplo, si queremos recorrer 5 centímetros a la derecha (eje x) y 3 centímetro hacia arriba (eje y) medidos desde el punto $(1,1)$, su código es `(1,1) ++(3,5)`.

Ejemplo 9.5

```
\begin{center}
  \begin{tikzpicture}
    \draw (0,0) grid (2,2);
    \draw[red] (1,0) -- ++(1,0) -- ++(0,1);
  \end{tikzpicture}
\end{center}
```

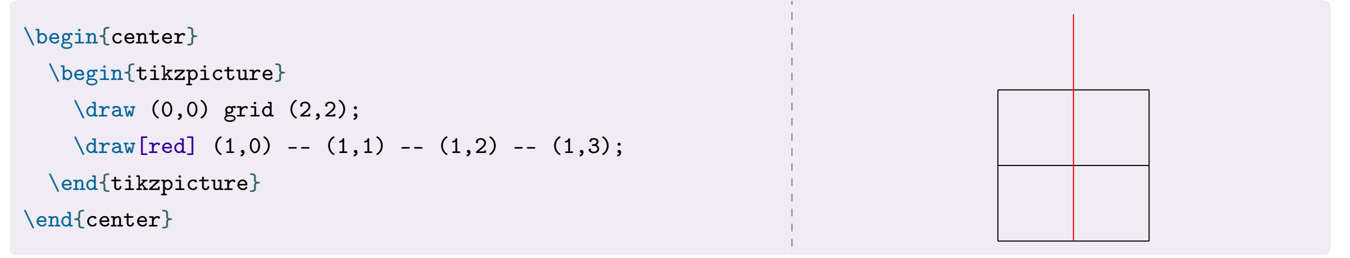
Los puntos de la forma `+(1,1)` también determinan un punto relativo, pero a diferencia del doble signo, este método no cambia el punto actual.

Ejemplo 9.6

```
\begin{center}
  \begin{tikzpicture}
    \draw (0,0) grid (2,2);
    \draw[red] (1,0) --+(1,0) --+(0,1);
  \end{tikzpicture}
\end{center}
```

Trazando una serie de líneas:

Ejemplo 9.7



```
\begin{center}
  \begin{tikzpicture}
    \draw (0,0) grid (2,2);
    \draw[red] (1,0) -- (1,1) -- (1,2) -- (1,3);
  \end{tikzpicture}
\end{center}
```

Para cerrar el trazado al se debe utilizar `cycle`:

Ejemplo 9.8

```
\begin{center}
  \begin{tikzpicture}
    \draw (0,0) grid (3,3);
    \draw[red] (0,0) -- (1,1) -- (2,1) -- (1,2) -- cycle;
  \end{tikzpicture}
\end{center}
```

Acciones con caminos

Un camino o un path es una serie de líneas rectas o curvas. Con estas líneas se pueden realizar ciertas acciones como cambiar el grosor, dibujar (draw) y entre otros.

```
draw
fill
shade
clip
```

Para definir un path se utiliza el comando `\path`. Por ejemplo,

Ejemplo 9.9

```
\begin{center}
  \begin{tikzpicture}
    \path (0,0) -- (1,0) -- (1,1) -- (0,1) -- cycle;
  \end{tikzpicture}
\end{center}
```

Los carácteres `--` sirven para conectar los puntos. Para dibujar vamos a utilizar `draw`,

Ejemplo 9.10

```
\begin{center}
  \begin{tikzpicture}
    \path[draw] (0,0) -- (1,0) -- (1,1) -- (0,1) -- cycle;
  \end{tikzpicture}
\end{center}
```

Para los demas casos:

Ejemplo 9.11

```
\begin{center}
  \begin{tikzpicture}
    \path[fill] (0,0) -- (1,0) -- (1,1) -- (0,1) -- cycle;
  \end{tikzpicture}
\end{center}
```

Ejemplo 9.12

```
\begin{center}
  \begin{tikzpicture}
    \path[shade] (0,0) -- (1,0) -- (1,1) -- (0,1) --
     ↪  cycle;
  \end{tikzpicture}
\end{center}
```

Ejemplo 9.13

```
\begin{center}
  \begin{tikzpicture}
    \path[fill] (-1,-1) -- (2,-1) -- (2,2) -- (-1,2) --
     ↪  cycle;
    \path[clip] (0,0) -- (1,0) -- (1,1) -- (0,1) -- cycle;
  \end{tikzpicture}
\end{center}
```

Ejemplo 9.14

```
\begin{center}
  \begin{tikzpicture}
    \path[clip] (0,0) -- (1,0) -- (1,1) -- (0,1) -- cycle;
    \path[fill] (-1,-1) -- (2,-1) -- (2,2) -- (-1,2) --
     ↪  cycle;
  \end{tikzpicture}
\end{center}
```

El clip simplemente hace un recortado.

Opciones de comandos tikz

Las opciones de los comandos `\tikz`, como también, del entorno `\tikzpicture`, son de tipo clave-valor (key-value), es decir, son de tipo `fill=red`.

Ejemplo 9.15

```
\begin{center}
  \begin{tikzpicture}
    \path[fill=Salmon!50!white] (-1,-1) -- (2,-1) -- (2,2)
    ↪  -- (-1,2) -- cycle;
  \end{tikzpicture}
\end{center}
```

El valor `Salmon!50!white` solo define el color y podemos reemplar por `red` o cualquier color.

En vez de las conexiones `--` se puede reemplazar por:

```
rectangle
circle
grid
```

Ejemplos:

Ejemplo 9.16

```
\begin{center}
  \begin{tikzpicture}
    \path[draw=Salmon, fill=Salmon!50!white] (0,0)
    ↪  rectangle (2,2);
  \end{tikzpicture}
\end{center}
```

Ejemplo 9.17

```
\begin{center}
  \begin{tikzpicture}
    \path[draw=Salmon, fill=Salmon!50!white] (0,0) circle
    ↪  (1cm);
  \end{tikzpicture}
\end{center}
```

Ejemplo 9.18

```
\begin{center}
  \begin{tikzpicture}
    \path[draw=Salmon] (0,0) grid (2,2);
  \end{tikzpicture}
\end{center}
```

Sintaxis especiales con nodo

Para crear un nodo (texto o cualquier objeto) se utiliza `node`,

Ejemplo 9.19

```
\begin{center}
  \begin{tikzpicture}
    \draw (0,0) node {texto} -- (2,2);
  \end{tikzpicture}
\end{center}
```

texto

Sintaxis del `node`

```
node[<opciones>]{<contenido>}
```

Estudiaremos las opciones de `node` más adelante.

Sintaxis especiales para crear árboles

Ejemplo 9.20

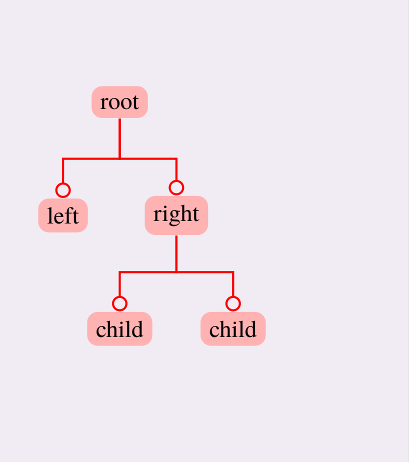
```
\begin{center}
  \begin{tikzpicture}
    \node {root}
      child {node {left}}
      child {node {right}
        child {node {left}}
        child {node {right}}
      };
  \end{tikzpicture}
\end{center}
```

root
left
right
left
right

El `child` genera una nueva rama, y dentro de estas ramas podemos generar otras ramas.

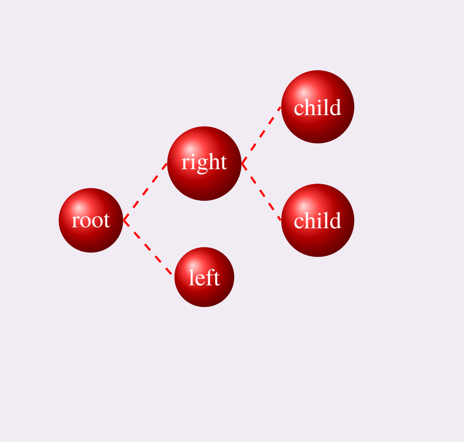
Ejemplo 9.21

```
\begin{center}
  \begin{tikzpicture}
    [edge from parent fork down, sibling distance=15mm,
    ↪  level distance=15mm,
    every node/.style={fill=red!30,rounded corners},
    edge from parent/.style={red,-o,thick,draw}]
    \node {root}
      child {node {left}}
      child {node {right}
        child {node {child}}
        child {node {child}}
      };
  \end{tikzpicture}
\end{center}
```



Ejemplo 9.22

```
\begin{center}
  \begin{tikzpicture}
    [parent anchor=east,child anchor=west,grow=east,
    sibling distance=15mm, level distance=15mm,
    every node/.style={ball color=red,circle,text=white},
    edge from parent/.style={draw,dashed,thick,red}]
    \node {root}
      child {node {left}}
      child {node {right}
        child {node {child}}
        child {node {child}}
      };
  \end{tikzpicture}
\end{center}
```



Sintaxis especiales con graph

El comando graph permite crear varios nodos.

Ejemplo 9.23

```
\begin{center}
  \begin{tikzpicture}
    \graph[grow down, branch right]{
      root -> {left, right -> {child, child}}
    };
  \end{tikzpicture}
\end{center}
```

root
left right
child

Entorno scope

El entorno scope permite crear una zona de trabajo, en donde a esta zona se le pueden aplicar ciertos operaciones o opciones.

Ejemplo 9.24

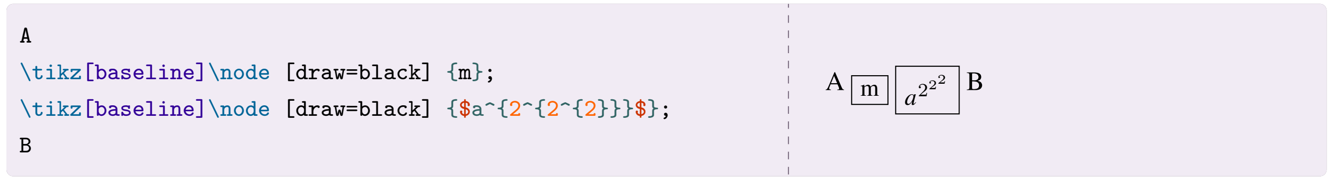
```
\begin{center}
  \begin{tikzpicture}
    \begin{scope}[color=red]
      \draw (0mm,10mm) -- (30mm,10mm);
      \draw (0mm, 8mm) -- (30mm, 8mm);
      \draw (0mm, 6mm) -- (30mm, 6mm);
    \end{scope}
    \begin{scope}[color=green]
      \draw (0mm, 4mm) -- (30mm, 4mm);
      \draw (0mm, 2mm) -- (30mm, 2mm);
      \draw[color=blue] (0mm, 0mm) -- (30mm, 0mm);
    \end{scope}
  \end{tikzpicture}
\end{center}
```

Si revisamos la documentación oficial de `tikz`, observaremos que muchas veces encontramos una cadena de la forma `/tikz/baseline`. Esta cadena nos indica que `baseline` es una opción de tikz y una cadena de la forma `/tcb/width` es una opción del `tcolorbox`.

Opción `/tikz/baseline`

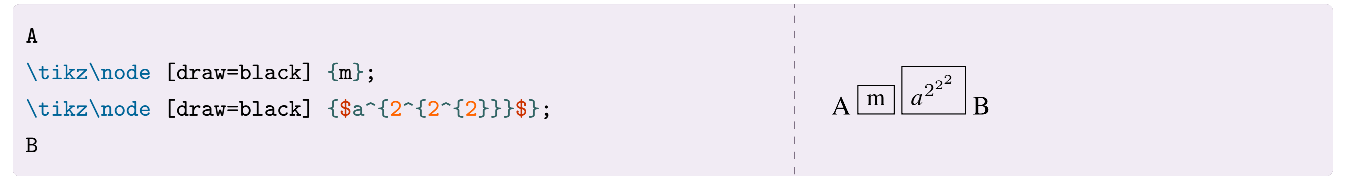
`/tikz/baseline=<dimension or coordinate>(default=0)` Esta opción alinea la gráfica con linea base.

Ejemplo 9.25



```
A
\tikz[baseline]\node [draw=black] {m};
\tikz[baseline]\node [draw=black] {$a^{2^{2^{2}}}$};
B
```


Ejemplo 9.26



```
A
\tikz\node [draw=black] {m};
\tikz\node [draw=black] {$a^{2^{2^{2}}}$};
B
```


Ejemplo 9.27

```
Hello \tikz[baseline=(X.base)] \node [cross out,draw] (X)
↪  {world};
```

Hello world

Uso de opciones de gráficas

Las opciones de tikz usan pgfkeys, por tanto, son de tipo `<key>=<value>`. Las opciones de tikz son de la forma `/tikz/<opción>`. Estas opciones podemos definir con el comando

```
\tikzet{<options>}
```

Uso y definición de estilos:

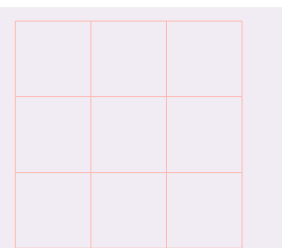
Ejemplo 9.28

```
\tikzset{mystyle/.style={Salmon!50!white}}
\begin{tikzpicture}[help lines/.style=mystyle]
  \draw[help lines] (0,0) grid (3,3);
\end{tikzpicture}
```

es equivalente a

Ejemplo 9.29

```
\begin{tikzpicture}[help lines/.style={Salmon!50!white}]
  \draw[help lines] (0,0) grid (3,3);
\end{tikzpicture}
```



También podemos realizar seteado global:

Ejemplo 9.30

```
\tikzset{help lines/.style={Salmon!50!white}}
\begin{center}
  \begin{tikzpicture}
    \draw[help lines] (0,0) grid (3,3);
  \end{tikzpicture}
\end{center}
```

Ejemplo 9.31

```
\tikzset{font=\small}
\tikzset{>=latex}
\begin{center}
  \begin{tikzpicture}
    \draw[->] (0,0) node{a} -- (1,1) node{b};
  \end{tikzpicture}
\end{center}
```

a b

El seteado global de las opciones de tikz se recomienda realizar en el preámbulo del documento LaTeX.

También podemos parametros para las opciones de tikz:

Ejemplo 9.32

```
\begin{center}
  \begin{tikzpicture}[%
      outline/.style={draw=#1,thick,fill=#1!50},
      outline/.default=Salmon]
    \node [outline=red] at (0,0) {red};
    \node [outline=blue] at (0,1) {blue};
    \node [outline] at (0,2) {default};
  \end{tikzpicture}
\end{center}
```

default

blue

red

Espesificacón de coordenadas

Coordenadas

Sintaxis para definir coordenadas:

```
([<options>]<coordinate>)
```

En las coordenadas podemos utilizar el sistema de coordenadas cartesianas, polares y esféricas.

Ejemplo 9.33

```
\begin{center}
  \begin{tikzpicture}
    \draw[help lines] (0,0) grid (3,3);
    \fill[blue] ([xshift=5mm,yshift=5mm]1,1) circle (3pt);
  \end{tikzpicture}
\end{center}
```

Coordenadas de nodo

Los nodos se ubican por medio de sus nombres.

Ejemplo 9.34

```
\begin{center}
  \begin{tikzpicture}
    \node (nombreNodoA) at (0,0) {A};
    \node[name=nombreNodoB] at (1,1) {B};
    \draw[->] (nombreNodoA) -- (nombreNodoB);
  \end{tikzpicture}
\end{center}
```

B
A

Coordenadas de intersecciones

La intersección de una línea vertical que pasa por el punto p y una línea horizontal que pasa por el punto q se determina por `(<p> |- <q>)` o por `(<q> -| <q>)`. Por ejemplo, `(2,1 |- 3,4)` y `(3,4 -| 2,1)` es equivalente al punto `(2,4)`.

Ejemplo 9.35

```
\begin{center}
  \begin{tikzpicture}
    \path (30:1cm) node(p1) {$p_1$} (75:1cm) node(p2)
    ↪  {$p_2$};
    \draw (-0.2,0) -- (1.2,0) node(xline)[right] {$q_1$};
    \draw (2,-0.2) -- (2,1.2) node(yline)[above] {$q_2$};
    \draw[->] (p1) -- (p1 |- xline);
    \draw[->] (p2) -- (p2 |- xline);
    \draw[->] (p1) -- (p1 -| yline);
    \draw[->] (p2) -- (p2 -| yline);
  \end{tikzpicture}
\end{center}
```

q_2
p_2
p_1
q_1

Para determinar las intersecciones de los paths se puede utilizar la librería `intersections`.

```
\usetikzlibrary{intersections}
```

Para determinar la intersección de 2 paths, primero debemos de ponerle nombres a los paths y esto se logra con los keys:

```
/tikz/name path = <name>
/tikz/name path global = <name>
```

Para que tikz encuentre las intercepciones de dos paths se debe utilizar el key,

```
/tikz/intersections path = <name>
```

Las intersecciones encontradas por este key se las nombran `intersection-1`, `intersection-2` y sucesivamente hasta n intersecciones.

```
/tikz/intersection/of=<name path 1> and <name path 2>
```

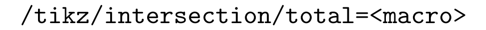
Este key especifica los nombres de los paths a utilizar para determinar las intersecciones.

```
/tikz/intersection/name=<prefix>(no default, initially intersection)
```

Este key especifica el prefijo de los nombres de las intersecciones.

Ejemplo 9.36

```
\begin{center}
  \begin{tikzpicture}[every node/.style={opacity=1, black, above left}]
      \draw [help lines] grid (3,2);
      \draw [name path=ellipse] (2,0.5) ellipse (0.75cm and 1cm);
      \draw [name path=rectangle, rotate=10] (0.5,0.5) rectangle +(2,1);
      \fill [red, opacity=0.5, name intersections={of=ellipse and rectangle}]
        (intersection-1) circle (2pt) node {1}
        (intersection-2) circle (2pt) node {2};
  \end{tikzpicture}
\end{center}
```

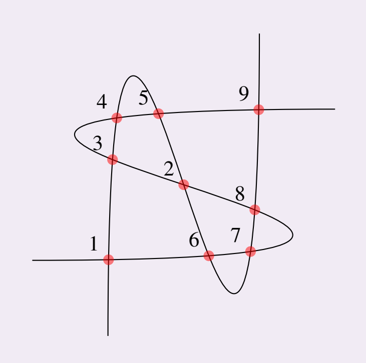


```
/tikz/intersection/total=<macro>
```

Este key determina el número total de intersecciones y luego asigna en un macro.

Ejemplo 9.37

```
\begin{center}
  \begin{tikzpicture}
    \clip (-2,-2) rectangle (2,2);
    \draw [name path=curve 1] (-2,-1) .. controls (8,-1) and (-8,1)
    ↪  .. (2,1);
    \draw [name path=curve 2] (-1,-2) .. controls (-1,8) and (1,-8)
    ↪  .. (1,2);
    \fill [name intersections={of=curve 1 and curve 2, name=i,
    ↪  total=\t}]
          [red, opacity=0.5, every node/.style={above left, black,
          ↪  opacity=1}]
          \foreach \s in {1,...,\t}{(i-\s) circle (2pt) node
          ↪  {\footnotesize\s}};
  \end{tikzpicture}
\end{center}
```

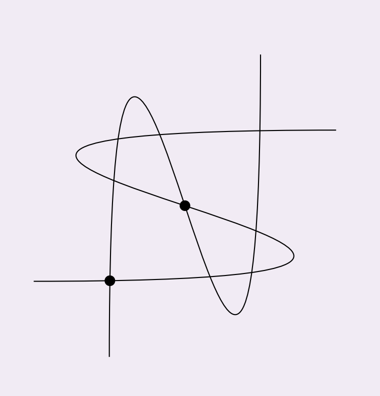



```
/tikz/intersection/by=<comma-separated list>
```

Este key nos permite especificar nombres de intersecciones.

Ejemplo 9.38

```
\begin{center}
  \begin{tikzpicture}
    \clip (-2,-2) rectangle (2,2);
    \draw [name path=curve 1] (-2,-1) .. controls (8,-1) and (-8,1)
    ↪  .. (2,1);
    \draw [name path=curve 2] (-1,-2) .. controls (-1,8) and (1,-8)
    ↪  .. (1,2);
    \fill [name intersections={of=curve 1 and curve 2, by={a,b}}]
          (a) circle (2pt)
          (b) circle (2pt);
  \end{tikzpicture}
\end{center}
```



Coordenadas relativas e incrementales

Debemos añadir ++ a una coordenada, para que este sea relativa.

Ejemplo 9.39

```
\begin{center}
  \begin{tikzpicture}
    \draw (0,0)     -- ++(1,0) -- ++(0,1) -- ++(-1,0) -- cycle;
    \draw (2,0)     -- ++(1,0) -- ++(0,1) -- ++(-1,0) -- cycle;
    \draw (1.5,1.5) -- ++(1,0) -- ++(0,1) -- ++(-1,0) -- cycle;
  \end{tikzpicture}
\end{center}
```

La nueva coordenada se establece como coordenada actual.

También podemos utilizar + para que este coordenada sea relativa con respecto al anterior, solo que a diferencia del ++, este no establece como un nuevo punto.

Ejemplo 9.40

```
\begin{center}
  \begin{tikzpicture}
    \draw (0,0) -- +(1,0) -- +(1,1) -- +(0,1) -- cycle;
    \draw[->] (1.5,1.5) -- +(30:1cm);
  \end{tikzpicture}
\end{center}
```

```
/tikz/turn
```

Este key nos determina un punto relativo, no solo con respecto al punto anterior, es con respecto a los dos puntos anteriores. Generalmente este key se utiliza como una opción de coordenada polar.

Ejemplo 9.41

```
\begin{center}
  \begin{tikzpicture}
    \draw (0,0) -- (1,1) -- ([turn]-45:1cm) --
    ↪ ([turn]-30:1cm);
  \end{tikzpicture}
\end{center}
```

Cáculo de coordenadas

Sintaxis general para calcular coordenadas:

```
([<options>]$<coordinate computation>$)
```

La sintaxis de factores de un punto:

```
<factor>*<coordinate><modifiers>
```

El factor puede ser cualquier expresión que puede revaluarse con `\pgfmathparse` y el factor se separa con $*$ de la parte restante. Si existe más de un $*$, tikz busca la última aparición de $*$.

Ejemplo 9.42

```
\begin{center}
  \begin{tikzpicture}
    \draw [help lines] (0,0) grid (3,2);
    \fill [red] ($2*(1,1)$) circle (2pt);
    \fill [green] (${1+1}*(1,.5)$) circle (2pt);
    \fill [blue] ($cos(0)*sin(90)*(1,1)$) circle (2pt);
    \fill [black] (${3*(4-3)}*(1,0.5)$) circle (2pt);
  \end{tikzpicture}
\end{center}
```

Sintaxis de modificadores parciales:

```
<coordinate>!<number>!<angle>:<second coordinate>
```

Ejemplo 9.43

```
\begin{center}
  \begin{tikzpicture}
    \draw [help lines] (0,0) grid (3,2);
    \draw (0,0) -- (3,2);
    \fill [red] ($(0,0)!0.5!(3,2)$) circle (2pt);
    \fill [blue] ($(0,0)!0.5!45:(3,2)$) circle (2pt);
  \end{tikzpicture}
\end{center}
```

Ejemplo 9.44

```
\begin{center}
  \begin{tikzpicture}
    \draw [help lines] (0,0) grid (3,2);
    \draw (0,0) -- (3,2);
    \draw[red] ($(0,0)!.3!(3,2)$) -- (3,0);
    \fill[red] ($(0,0)!.3!(3,2)!.7!(3,0)$) circle (2pt);
  \end{tikzpicture}
\end{center}
```

Sintaxis de modificadores de distancia:

```
<coordinate>!<dimension>!<angle>:<second coordinate>
```

Ejemplo 9.45

```
\begin{center}
  \begin{tikzpicture}
    \draw [help lines] (0,0) grid (3,2);
    \draw (0,0) -- (3,2);
    \fill [red] ($(0,0)!1cm!(3,2)$) circle (2pt);
    \fill [blue] ($(0,0)!1cm!45:(3,2)$) circle (2pt);
  \end{tikzpicture}
\end{center}
```

Ejemplo 9.46

```
\begin{center}
  \begin{tikzpicture}
    \draw [help lines] (0,0) grid (3,2);
    \coordinate (a) at (1,0);
    \coordinate (b) at (3,1);
    \draw (a) -- (b);
    \coordinate (c) at ($ (a)!.25!(b) $);
    \coordinate (d) at ($ (c)!1cm!90:(b) $);
    \draw [<->] (c) -- (d) node [sloped,midway,above]
    ↪  {1cm};
  \end{tikzpicture}
\end{center}
```

1cm

Sintaxis de modificadores de proyección:

```
<coordinate>!<proyection coordinate>!<angle>:<second coordinate>
```

Ejemplo 9.47

```
\begin{center}
  \begin{tikzpicture}
    \draw[help lines] (0,0) grid (3,2);
    \coordinate (a) at (0,1);
    \coordinate (b) at (3,2);
    \coordinate (c) at (2.5,0);
    \draw (a) -- (b) -- (c) -- cycle;
    \draw[red](a) -- ($(b)!(a)!(c)$);
    \draw[orange] (b) -- ($(a)!(b)!(c)$);
    \draw[blue] (c) -- ($(a)!(c)!(b)$);
  \end{tikzpicture}
\end{center}
```

Especificación de un path

```
\path<specification>;
```

Opciones:

1. Esquinas redondeadas y punteadas:

Ejemplo 9.48

```
\begin{center}
  \begin{tikzpicture}
    \draw (0,0) -- (1,0)
    [rounded corners] -- (1,1) -- (2,3)
    [sharp corners] -- (2,2) -- (0,2);
  \end{tikzpicture}
\end{center}
```

2. Efecto de un key sobre un path, "scoped":

Ejemplo 9.49

```
\begin{center}
  \begin{tikzpicture}
    \draw (0,0) -- (1,0)
    {[rounded corners] -- (1,1) -- (2,3)}
    -- (2,2) -- (0,2);
  \end{tikzpicture}
\end{center}
```

De manera análoga podemos cambiar los colores de los trazos:



Ejemplo 9.50

```
\begin{center}
  \begin{tikzpicture}
    \draw (0,0) -- (1,1)
    [color=red] -- (2,0) -- (3,1)
    [color=blue] -- (3,0) -- (2,1);
  \end{tikzpicture}
\end{center}
```

Ejemplo 9.51

```
\begin{center}
  \begin{tikzpicture}
    \draw (0,0) -- (1,1);
    \draw [color=red] -- (2,0) -- (3,1);
    \draw [blue] (3,0) -- (2,1);
  \end{tikzpicture}
\end{center}
```

```
/tikz/name = <name path>
```

Este key asigna un nombre a un path.

```
/tikz/every path
```

Este key establece el estilo del path.

Ejemplo 9.52

```
\begin{center}
  \begin{tikzpicture}
    [fill=yellow!80!black, % only sets the color
    every path/.style={draw}]  % all paths are drawn
    \fill  (0,0) rectangle +(1,1);
    \shade (2,0) rectangle +(1,1);
  \end{tikzpicture}
\end{center}
```

```
/tikz/insert path = <path>
```

Este key se utiliza para insertar algo en un path.

Ejemplo 9.53

```
\begin{center}
  \begin{tikzpicture}[c/.style={insert
  ↪  path={circle[radius=2pt]}}]
    \draw (0,0) -- (1,1) [c] -- (3,2) [c];
  \end{tikzpicture}
\end{center}
```

La operación de mover:

```
\path ... <coordinate> ...;
```

Es utilizado para cortar o mover un path.

Ejemplo 9.54

```
\begin{center}
  \begin{tikzpicture}
    \draw (0,0) --(2,0) (0,1) --(2,1);
  \end{tikzpicture}
\end{center}
```

Operación de la línea:

```
\path ... -- <coordinate> ...;
```

Ejemplo 9.55

```
\begin{center}
  \begin{tikzpicture}
    \draw (0,0) -- (2,0);
  \end{tikzpicture}
\end{center}
```

Líneas horizontales y verticales:

```
\path ... -| <coordinate or cycle> ...;
\path ... |- <coordinate or cycle> ...;
```

La primera forma nos indica que se debe trazar primeramente la línea horizontal y luego vertical. La segunda forma nos indica que se debe de trazar primeramente la línea vertical y luego la horizontal.

Ejemplo 9.56

```
\begin{center}
  \begin{tikzpicture}
    \draw (0,0) node(a) [draw] {A}  (1,1) node(b) [draw]
    ↪  {B};
    \draw (a.north) |- (b.west);
    \draw[color=red] (a.east) -| (2,1.5) -| (b.north);
  \end{tikzpicture}
\end{center}
```


Curva:

```
\path ... ..controls<c>and<d>..<y or cycle> ...;
```

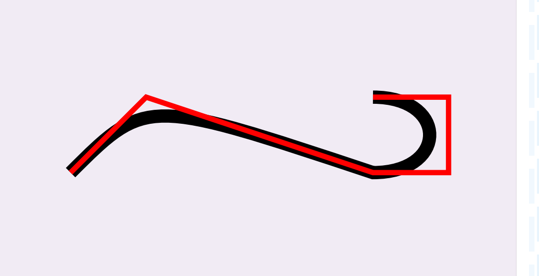
Supongamos que tenemos el código:

```
(4,0) .. controls (5,0) and (5,1) .. (4,1);
```

El path se extiende desde (4,0) hasta (4,1). La curva comienza de (4,0) con dirección a (5,0) (la recta tangente de la curva que pasa por el punto (4,0) incluye al punto (5,0)) y termina en punto (4,1) con punto de control (5,1). Si no se especifica d, entonces, d adopta el valor de c. La curva es una curva de Bézier con puntos de control c y d.

Ejemplo 9.57

```
\begin{center}
  \begin{tikzpicture}
    \draw[line width=5pt] (0,0) .. controls (1,1) .. (4,0)
    ↪  .. controls (5,0) and (5,1) .. (4,1);
    \draw[color=red, line width=2pt] (0,0) -- (1,1) --
    ↪  (4,0) -- (5,0) -- (5,1) -- (4,1);
  \end{tikzpicture}
\end{center}
```



Operación de rectángulo:

```
\path ... rectangle <corner or cycle> ...;
```

Este operación representa un rectángulo.

Ejemplo 9.58

```
\begin{center}
  \begin{tikzpicture}
    \draw (0,0) rectangle (1,1);
  \end{tikzpicture}
\end{center}
```

Esquinas redondeadas:

```
/tikz/rounded corners = <inset> (default 4pt)
```

Ejemplo 9.59

```
\begin{center}
  \begin{tikzpicture}
    \draw[rounded corners=4pt] (0,0) rectangle (3,1);
  \end{tikzpicture}
\end{center}
```

Esquinas punteadas:

```
/tikz/sharp corners
```

Ejemplo 9.60

```
\begin{center}
  \begin{tikzpicture}
    \draw[sharp corners] (0,0) rectangle (3,1);
  \end{tikzpicture}
\end{center}
```

Generalmente este key es utilizado cuando se tiene líneas con esquinas redondeadas.

Operación de círculo y elipse:

```
\path ... circle[<options>] ...;
\path ... ellipse[<options>] ...;
```

Opciones:

1. `/tikz/radius=<value>`. Este key define el radio del círculo.
2. `/tikz/x radius=<value>`. Este key define el radio en el eje x del elipse.
3. `/tikz/y radius=<value>`. Este key define el radio en el eje y del elipse.
4. `/tikz/at=<coordinate>`. Este key define la posición del centro del círculo o elipse.

Ejemplo 9.61

```
\begin{center}
  \begin{tikzpicture}
    \draw (3,3) circle [radius=2cm];
    \draw (3,3) circle [x radius=2cm, y radius=1cm];
  \end{tikzpicture}
\end{center}
```

Arco:

```
\path ... arc[<options>] ...;
```

Opciones:

1. `/tikz/start angle=<degrees>`. Este key define el ángulo de inicio del arco.
2. `/tikz/end angle=<degrees>`. Este key define el ángulo final del arco.
3. `/tikz/delta angle=<degrees>`. Este key define el ángulo de separación entre los segmentos del arco.

También se debe de utilizar el key radius para definir el radio del arco.

Ejemplo 9.62

```
\begin{center}
  \begin{tikzpicture}
    \draw[red] (0,0) -- +(30:1cm) arc [radius=1cm, start
    ↪  angle=30,  delta angle=30] -- cycle;
    \draw[blue] (2,0) -- +(30:1cm) arc [radius=1cm, start
    ↪  angle=30,  end angle=60] -- cycle;
  \end{tikzpicture}
\end{center}
```

Ejemplo 9.63

```
\begin{center}
  \begin{tikzpicture}[radius=1cm,delta angle=30]
    \draw (-1,0) -- +(3.5,0);
    \draw (1,0) ++(210:2cm) -- +(30:4cm);
    \draw (1,0) +(0:1cm) arc [start angle=0];
    \draw (1,0) +(180:1cm) arc [start angle=180];
    \path (1,0) ++(15:.75cm) node{$\alpha$};
    \path (1,0) ++(15:-.75cm) node{$\beta$};
  \end{tikzpicture}
\end{center}
```

Otra forma para realizar arcos es:

```
\path ... (<start angle>:<end angle>:<radius>) ...;
```

Ejemplo 9.64

```
\begin{center}
  \begin{tikzpicture}
    \draw (0,0) -- (1,0) arc (0:45:1cm) -- cycle;
  \end{tikzpicture}
\end{center}
```

Operación grid o cuadrícula:

```
\path ... grid[<options>]<corner or cycle> ...;
```

Opciones:

1. /tikz/step=<number or dimension or coordinate>(default 1cm). Este key define el tamaño de la cuadrícula en dirección x y y.
2. /tikz/xstep=<dimension or number>. Este key define el tamaño de la cuadrícula en dirección x.

3. `/tikz/ystep=<dimension or number>`. Este key define el tamaño de la cuadrícula en dirección y.

4. `help lines`. Este key define si se deben de dibujar las líneas en color gris (de ayuda).

Ejemplo 9.65

```
\begin{center}
  \begin{tikzpicture}
    \draw (0,0) grid (3,3);
  \end{tikzpicture}
\end{center}
```

Ejemplo 9.66

```
\begin{center}
  \begin{tikzpicture}
    \draw (0,0) grid[xstep=0.5cm,ystep=0.5cm] (3,3);
  \end{tikzpicture}
\end{center}
```

Ejemplo 9.67

```
\begin{center}
  \begin{tikzpicture}
    \draw (0,0) grid[xstep=1cm,ystep=1cm,help lines]
    ↪  (3,3);
  \end{tikzpicture}
\end{center}
```

En la documentación oficial de tikz existe la operación de parábola, sino y coseno, que realizan las gráficas de estes funciones. En este libro no desarrollaremos estas operaciones debido a que estaremos realizando estas gráficas con el paquete `pgfplots`.

Operación de ruta de acceso:

```
\path ... at[<options>] <nodes> <coordinate or cycle> ...;
```

Esta operación es similar que `--`, solo que en este caso podemos especificar el ángulo de salida del path y el ángulo de entrada.

Ejemplo 9.68

```
\begin{center}
  \begin{tikzpicture}
    \draw (0,0) to[out=90,in=145] (2,2);
  \end{tikzpicture}
\end{center}
```

Ejemplo 9.69

```
\begin{center}
  \begin{tikzpicture}
    \draw (0,0) to[out=90,in=145] node [sloped,above]
    ↪ {$s(t)$} (2,2);
  \end{tikzpicture}
\end{center}
```

$s(t)$

Opciones:

1. `/tikz/edge node=<node specification>`. Este key agrega nodos a lo largo del path (conexión).

Ejemplo 9.70

```
\begin{center}
  \begin{tikzpicture}
    \draw[-latex] (0,0) to [edge node={node
    ↪ [sloped,above]{$\vec{x}$}}] (2,2);
  \end{tikzpicture}
\end{center}
```

$\vec{x}$

2. `/tikz/edge label=<text>`. Este key agrega una etiqueta a lo largo del path.

Ejemplo 9.71

```
\begin{center}
  \begin{tikzpicture}
    \draw (0,0) to [edge label={$\vec{x}$}] (2,2);
  \end{tikzpicture}
\end{center}
```

$\vec{x}$

3. `/tikz/edge label'=<text>`.

Ejemplo 9.72

```
\begin{center}
  \begin{tikzpicture}
    \draw (0,0) to [edge label'={$\vec{x}$}] (2,2);
  \end{tikzpicture}
\end{center}
```

4. /tikz/every to.

Ejemplo 9.73

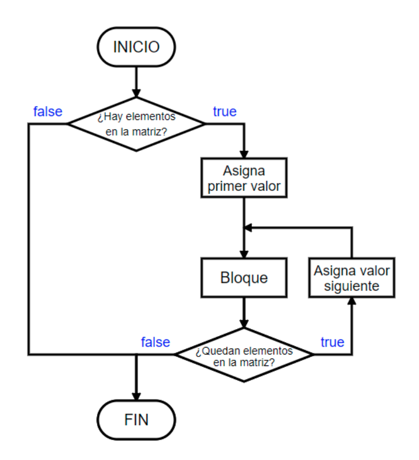
```
\begin{center}
  \begin{tikzpicture}[every to/.style={bend left}]
    \draw (0,0) to (2,2);
  \end{tikzpicture}
\end{center}
```

Operación foreach:

```
\path ... foreach<variables>[<options>] in {<path commands>} ...;
```

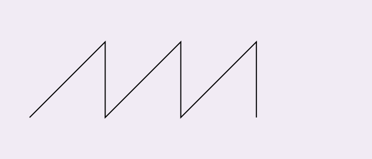
Esta operación es un bucle.

Figura 9.1



Ejemplo 9.74

```
\begin{center}
  \begin{tikzpicture}
    \draw (0,0) foreach \x in {1,...,3}{-- (\x,1) --
    ↪  (\x,0)};
  \end{tikzpicture}
\end{center}
```



Acciones con path

Una vez construido un path, se puede realizar diferentes acciones con este `path`, como, por ejemplo, trazar, rellenar y sombrear. También existen otras acciones que estaremos estudiando a detalle.

Se pueden aplicar acciones a un `path` por medio de dos métodos. El primer método consiste en utilizar los comandos como `\draw`, `\fill` y entre otros. El segundo método consiste en utilizar keys como `draw`, `fill` y entre otros. El comando `\path` sirve para especificar una ruta o camino.

Lista de acciones:

1. `\draw` es equivalente a `\path[draw]`.
2. `\fill` es equivalente a `\path[fill]`.
3. `\filldraw` es equivalente a `\path[fill,draw]`.
4. `\pattern` es equivalente a `\path[pattern]`.
5. `\shade` es equivalente a `\path[shade]`.
6. `\shadedraw` es equivalente a `\path[shade,draw]`.
7. `\clip` es equivalente a `\path[clip]`.
8. `\useasboundingbox` es equivalente a `\path[use as bounding box]`.

Especificación de color para las acciones `\draw` y `\fill`:

```
/tikz/color=<color name>
```

Ejemplo 9.75

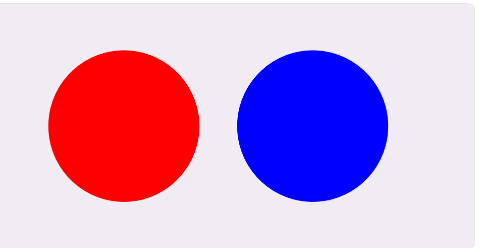
```
\begin{center}
  \begin{tikzpicture}
    \draw[red] (0,0) circle (1cm);
    \draw[color=blue] (2.5,0) circle (1cm);
  \end{tikzpicture}
\end{center}
```

Ejemplo 9.76

```
\begin{center}
  \begin{tikzpicture}
    \path[draw,red] (0,0) circle (1cm);
    \path[draw,color=blue] (2.5,0) circle (1cm);
  \end{tikzpicture}
\end{center}
```

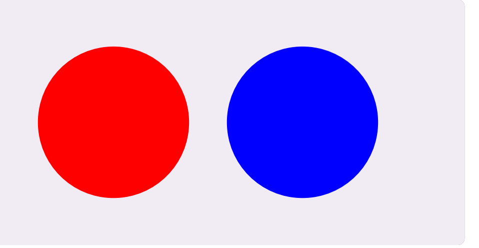
Ejemplo 9.77

```
\begin{center}
  \begin{tikzpicture}
    \fill[red] (0,0) circle (1cm);
    \fill[color=blue] (2.5,0) circle (1cm);
  \end{tikzpicture}
\end{center}
```



Ejemplo 9.78

```
\begin{center}
  \begin{tikzpicture}
    \path[fill,red] (0,0) circle (1);
    \path[fill,color=blue] (2.5,0) circle (1cm);
  \end{tikzpicture}
\end{center}
```



Dibujado de trazados:

```
/tikz/draw=<color>
```

Ejemplo 9.79

```
\begin{center}
  \begin{tikzpicture}
    \path[draw=red] (0,0) -- (1,1) circle (1cm);
  \end{tikzpicture}
\end{center}
```

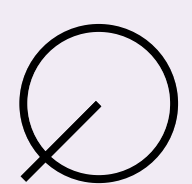


Ancho de línea:

```
/tikz/line width=<dimension>
```

Ejemplo 9.80

```
\begin{center}
  \begin{tikzpicture}
    \draw[line width=3pt] (0,0) -- (1,1) circle (1cm);
  \end{tikzpicture}
\end{center}
```



Ancho de líneas predefinidos:

1. `/tikz/ultra thin`. Es de valor $0{,}1$ pt.

Ejemplo 9.81

```
\begin{center}
  \begin{tikzpicture}
    \draw[ultra thin] (0,0) -- (1,1) circle (1);
  \end{tikzpicture}
\end{center}
```

2. `/tikz/very thin`. Es de valor 0,2 pt.
3. `/tikz/thin`. Es de valor 0,4 pt.
4. `/tikz/semithick`. Es de valor 0,6 pt.
5. `/tikz/thick`. Es de valor 0,8 pt.
6. `/tikz/very thick`. Es de valor 1,2 pt.
7. `/tikz/ultra thick`. Es de valor 1,6 pt.

Estilos de inicio y finalizado de líneas:

```
/tikz/line cap=<type>
```

Ejemplo 9.82

```
\begin{center}
  \begin{tikzpicture}[line width=10pt]
    \draw[line cap=round] (0,2) -- ++(2,0);
    \draw[line cap=butt] (0,1) -- ++(2,0);
    \draw[line cap=rect] (0,0) -- ++(2,0);
    \draw (0,-1) -- ++(2,0);
  \end{tikzpicture}
\end{center}
```

Unión de líneas:

```
/tikz/line join=<type>
```

Ejemplo 9.83

```
\begin{center}
  \begin{tikzpicture}[line width=10pt]
    \draw[line join=round] (0,0) -- ++(.5,1) -- ++(.5,-1);
    \draw[line join=bevel] (2,0) -- ++(.5,1) -- ++(.5,-1);
    \draw[line join=miter] (4,0) -- ++(.5,1) -- ++(.5,-1);
  \end{tikzpicture}
\end{center}
```



Lineas descuntinuas:

1. /tikz/dash pattern=<dash pattern>. Lleva la misma sintaxis que metafont.

Ejemplo 9.84

```
\begin{center}
  \begin{tikzpicture}
    \draw[dash pattern=on 2pt off 3pt on 4pt off 4pt]
    ↪  (0,0) -- (3,0);
  \end{tikzpicture}
\end{center}
```

2. /tikz/dash phase=<dash phase>.
3. /tikz/solid.
4. /tikz/dotted.
5. /tikz/densely dotted.
6. /tikz/loosely dotted.
7. /tikz/dashed.

Ejemplo 9.85

```
\begin{center}
  \begin{tikzpicture}
    \draw[dashed] (0,0) -- (3,0);
  \end{tikzpicture}
\end{center}
```

8. /tikz/densely dashed.
9. /tikz/loosely dashed.
10. /tikz/dash dot.
11. /tikz/densely dash dot.
12. /tikz/loosely dash dot.
13. /tikz/dash dot dot.
14. /tikz/densely dash dot dot.
15. /tikz/loosely dash dot dot.

Líneas dobles:

```
/tikz/double=<core color(default white)>
```

Este key genera doble línea.

Ejemplo 9.86

```
\begin{center}
  \tikz\draw[double=red] (0,0) -- (3,1);
\end{center}
```

```
/tikz/double distance=<dimension (default 0.6pt)>
```

Este key estable la distancia de separación de doble línea.

Ejemplo 9.87

```
\begin{center}
  \tikz\draw[double distance=1cm] (0,0) -- (3,1);
\end{center}
```

Para rellenar un trazado se debe de recurrir al key

```
/tikz/fill=<color>
```

Para que se pueda rellenar un `\path` es necesario que esté cerrado, y si no está cerrado, automáticamente se cierra.

Ejemplo 9.88

```
\begin{center}
  \begin{tikzpicture}
    \path[draw, fill=black!20!white] (0,0) -- ++(2,1) --
    ↪  ++(0,-1) -- cycle;
    \draw[fill] (2.5,0) -- ++(2,1) -- ++(0,-1);
    \fill (1,-1.5) -- ++(2,1) -- ++ (0,-1);
  \end{tikzpicture}
\end{center}
```

Relleno con patrones:

```
/tikz/pattern=<name>
```

Para rellenar con patrones, es necesario importar la librería `pattern`.

```
\usetikzlibrary{pattern}
```

Ejemplo 9.89

```
\begin{center}
  \begin{tikzpicture}
    \draw[pattern=dots] (0,0) circle (1.5cm);
  \end{tikzpicture}
\end{center}
```

Especificación de color de un patron:

```
/tikz/pattern color=<color>
```

Ejemplo 9.90

```
\begin{center}
  \begin{tikzpicture}
    \draw[pattern=dots, pattern color=red] (0,0) circle
    ↪  (1.5cm);
  \end{tikzpicture}
\end{center}
```

Sombreado de un trazado:

```
/tikz/shade
```

Es algo similar que el key de relleno.

Ejemplo 9.91

```
\begin{center}
  \begin{tikzpicture}
    \begin{scope}
      \path[draw,shade] (0,0) circle (0.5cm);
    \end{scope}
    \begin{scope}[yshift=-1.2cm]
      \shade (0,0) circle (0.5cm);
    \end{scope}
    \begin{scope}[yshift=-2.4cm]
      \shadedraw (0,0) circle (0.5cm);
    \end{scope}
  \end{tikzpicture}
\end{center}
```

Como podemos apreciar, el sombreado predeterminado es la transición de un color gris a blanco, de arriba a abajo. Es posible cambiar estos parámetros como el color de sombreado.

```
/tikz/shading=<name>
```

Las opciones existentes son: axis, radial y ball.

Ejemplo 9.92

```
\begin{center}
  \begin{tikzpicture}
    \begin{scope}
      \path[draw,shading=axis] (0,0) rectangle (2,0.5);
    \end{scope}
    \begin{scope}[yshift=-1.2cm]
      \shade[draw, shading=radial] (0,0) rectangle
       ↪ (2,0.5);
    \end{scope}
    \begin{scope}[yshift=-2.4cm]
      \shadedraw[shading=ball] (0,0) circle (0.5cm);
    \end{scope}
  \end{tikzpicture}
\end{center}
```

Cambiando el color:

Ejemplo 9.93

```
\begin{center}
  \begin{tikzpicture}
    \draw[shade, left color=red, right color=white] (0,0)
     ↪ rectangle (2,2);
  \end{tikzpicture}
\end{center}
```

Ángulo de sombreado:

```
/tikz/shading angle=<angle> default 0
```

Ejemplo 9.94

```
\begin{center}
  \begin{tikzpicture}
    \draw[shade, shading angle=-45] (0,0) rectangle (2,2);
  \end{tikzpicture}
\end{center}
```

Recortado por medio de un trazado:

```
/tikz/clip
```

Este key restringe el área de dibujado, este trazado de restricción puede ser un rectángulo, circulo o una ruta cualquiera.

Ejemplo 9.95

```
\begin{center}
  \begin{tikzpicture}
    \draw[clip] (0,0) circle (1.5cm);
    \fill[Salmon!50] (1,0) circle (1.5cm);
  \end{tikzpicture}
\end{center}
```

También se puede utilizar el comando `\clip` en vez de `\path[clip]`.

Ejemplo 9.96

```
\begin{center}
  \begin{tikzpicture}
    \clip (0,0) circle (1.5cm);
    \fill[Salmon!50] (1,0) circle (1.5cm);
  \end{tikzpicture}
\end{center}
```

Todos los trazados después del `\clip` serán recortados.

Para decorar un `\path` es necesario importar la librería `decorations`

Ejemplo 9.97

```
\begin{center}
  \begin{tikzpicture}
    \draw[decorate, decoration=zigzag] (0,0) -- (3,0);
  \end{tikzpicture}
\end{center}
```

Arrows o flechas

Una línea o curva tiene un punto inicial (donde inicia la línea o la curva) y un punto final (donde termina la línea o la curva). Tanto en el punto inicial o en el final se pueden colocar flechas. Para colocar una flecha, simplemente debemos utilizar `->` (también puede ser `<-` y además podemos reemplazar > por un parámetro definido) como un argumento de un `\path`.

Ejemplo 9.98

```
\begin{center}
  \begin{tikzpicture}
    \draw[->] (0,0) -- (3,0);
  \end{tikzpicture}
\end{center}
```

Ejemplo 9.99

```
\begin{center}
  \begin{tikzpicture}
    \draw[<-] (0,0) -- (3,0);
  \end{tikzpicture}
\end{center}
```

Ejemplo 9.100

```
\begin{center}
  \begin{tikzpicture}
    \draw[latex-latex] (0,0) -- (3,0);
  \end{tikzpicture}
\end{center}
```

Para añadir una flecha, también podemos utilizar el siguiente key:

```
/tikz/arros=<start arrow>-<end arrow>
```

Ejemplo 9.101

```
\begin{center}
  \begin{tikzpicture}
    \draw[arrows=-latex] (0,0) -- (3,0);
  \end{tikzpicture}
\end{center}
```

Tamaño de una flecha:

```
/pgf/arrow key/lenght=<dimension><line width><outer factor>
```

Ejemplo 9.102

```
\begin{center}
  \begin{tikzpicture}
    \draw [-{Stealth[length=5mm]}] (0,0) -- (2,0);
    \draw [|<->|] (1.5,.4) -- node[above=1mm] {5mm} (2,.4);
  \end{tikzpicture}
\end{center}
```

5mm

Ejemplo 9.103

```
\begin{center}
  \begin{tikzpicture}
    \draw[line width=1pt, double distance=3pt, arrows = {-Latex[length=0pt 3
    ↪  0]}] (0,0) -- (1,0);
  \end{tikzpicture}
\end{center}
```

Ancho de la punta de flecha:

```
/pgf/arrow keys/width=<dimension><line width factor><outer factor>
```

Ejemplo 9.104

```
\begin{center}
  \begin{tikzpicture}
    \draw[arrows = {-Latex[width=10pt, length=10pt]}] (0,0) -- (1,0);
  \end{tikzpicture}
\end{center}
```

Parámetro de una flecha sigilosa:

```
/pgf/arrow keys/inset=<dimension><line width factor><outer factor>
```

Ejemplo 9.105

```
\begin{center}
  \begin{tikzpicture}
    \draw[arrows = {-Stealth[length=10pt, inset=2pt]}] (0,0) -- (1,0);
  \end{tikzpicture}
\end{center}
```

Flecha definida a partir de un ángulo y ancho:

```
/pgf/arrow keys/angle=<angle>:<dimension><line widht factor><outer factor>
```

Ejemplo 9.106

```
\begin{center}
  \begin{tikzpicture}
    \draw [arrows = {-Stealth[angle=30:8pt]}] (0,0) -- (1,0);
  \end{tikzpicture}
\end{center}
```

La punta de la flecha está a un ángulo de 30 grados.

Flechas al reveresa:

```
/pgf/arrow keys/reversed
```

Este key sirve para poner en reversa la dirección de la flecha.

Ejemplo 9.107

```
\begin{center}
  \begin{tikzpicture}
    \draw [arrows = {-Stealth[reversed]}] (0,0) -- (1,0);
  \end{tikzpicture}
\end{center}
```

Flechas mitades:

```
/pgf/arrow keys/harpoon
```

Con este key se obtiene la mitad izquierdo de la flecha.

Ejemplo 9.108

```
\begin{center}
  \begin{tikzpicture}
    \draw [arrows = {-Stealth[harpoon]}] (0,0) -- (1,0);
  \end{tikzpicture}
\end{center}
```

Para obtener la otra mitad debemos utilizar el key,

```
/pgf/arrow keys/swap
```

Ejemplo 9.109

```
\begin{center}
  \begin{tikzpicture}
    \draw [arrows = {-Stealth[swap, harpoon]}] (0,0) --
    ↪ (1,0);
  \end{tikzpicture}
\end{center}
```

Apariencias de las diferentes flechas:

Nodes and Edges

En tikz, se la denomina modo a cualquier rectángulo, círculo otra forma (en algunas ocasiones hasta suele denominarse nodo a un punto del sistema de coordenadas cartesianas), como también puede ser un texto. Un nodo básicamente puede tener altura, ancho, color y nombre.

Sintaxis para especificar un nodo:

```
\path ... node<foreach statements>[<options>](<name>) at (<coordinate>){<node
↪  contents>}...;
```

Contenido de un nodo:

```
/tikz/node contents=<node contents>
```

Ejemplo 9.110

```
\begin{center}
  \begin{tikzpicture}
    \node[red] (a) at (0,0) {A};
    \path (1,0) node [blue, name=b] {B};
    \path (2,0) node [green, node contents=D, name=d];
  \end{tikzpicture}
\end{center}
```

A B D

Ubicación del nodo:

```
/tikz/at=<coordinate>
```

Este que sirve para ubicar un modo en el sistema cartesiano por mediante especificaciones de coordenadas.

Dibujando un nodo detrás de un `path`:

```
/tikz/behind path
```

Ejemplo 9.111

```
\begin{center}
  \begin{tikzpicture}
    \fill[green]
    (0,0) node [behind path, draw, fill=red] {A}
    --(2,0) node [behind path, draw, fill=blue] {B}
    --(2,2) node [behind path, draw, fill=Salmon!50] {C};
  \end{tikzpicture}
\end{center}
```

C
A B

Dibujando un node delante de un `path`:

```
/tikz/in front of path
```

Ejemplo 9.112

```
\begin{center}
  \begin{tikzpicture}
    \fill[green]
    (0,0) node [in front of path, draw, fill=red] {A}
    --(2,0) node [in front of path, draw, fill=blue] {B}
    --(2,2) node [in front of path, draw, fill=Salmon!50] {C};
  \end{tikzpicture}
\end{center}
```

C
A B

Asignando un nombre a un nodo:

```
/tikz/name=<name>
```

Ejemplo 9.113

```
\begin{center}
  \begin{tikzpicture}
    \node[red, name=a] at (0,0) {A};
    \node[red] (b) at (2,2) {B};
    \draw (a) -- (b);
  \end{tikzpicture}
\end{center}
```

B
A

Asignar otro nombre al nodo:

```
/tikz/alias=<another node name>
```

Ejemplo 9.114

```
\begin{center}
  \begin{tikzpicture}
    \node[red, alias=a] at (0,0) {A};
    \node[red] (b) at (2,2) {B};
    \draw (a) -- (b);
  \end{tikzpicture}
\end{center}
```

B
A

Los nodos en la mayoría de los casos aceptan opciones de acciones de un path, pero no todos; por ejemplo,

Ejemplo 9.115

```
\begin{center}
  \begin{tikzpicture}
    \path[fill=Salmon!50!white] (0,0) rectangle (4,2);
    \node[fill=white, draw=Salmon, double, rounded
    ↪  corners=2pt] at (2,1) {Heber MQ};
  \end{tikzpicture}
\end{center}
```

Heber MQ

La forma por defecto que tiene un node es rectángulo,

Ejemplo 9.116

```
\tikz\node[draw]{Heber};
```

Heber

pero esta forma se puede cambiar con el key

```
/tikz/shape=<shape name>
```

Las diferentes formas están establecidas en la librería `shapes`, entre ellos tenemos `circle`.

Ejemplo 9.117

```
\tikz\node[draw, shape=circle]{Heber};
```

Heber

Nodos con foreach:

Ejemplo 9.118

```
\tikz \draw (0,0) node foreach \x in {1,2,3} at (\x,0)
↪  {\x};
```

1 2 3

Estilo de un nodo:

```
/tikz/every node
```

Ejemplo 9.119

```
\begin{center}
  \begin{tikzpicture}[every
  ↪  node/.style={fill=Salmon!50!white, rounded
  ↪  corners=2pt}]
    \draw (0,0) node {A} -- (2,1) node {B};
  \end{tikzpicture}
\end{center}
```

B
A

Personalizar estilo de una forma de un nodo:

```
/tikz/every <shape> node
```

Ejemplo 9.120

```
\begin{center}
  \begin{tikzpicture}[%
      every node/.style={fill=Salmon!50!white}
      every rectangle node/.style={draw, rounded corners=2pt}
      every circle node/.style={draw, double}%
    ]
    \draw (0,0) node[rectangle] {A} -- (2,1) node[circle] {B};
  \end{tikzpicture}
\end{center}
```


Formas predefinidos por `tikz`:

1. `rectangle`
2. `circle`
3. `coordinate`

Para su uso, simplemente debemos de utilizar una de las opciones enumeradas como un valor de key `shape`.

Ejemplo 9.121

```
\begin{center}
  \begin{tikzpicture}
    \node[shape=rectangle, draw] at (0,0) {Heber MQ};
    \node[shape=circle, draw] at (2,0) {Heber MQ};
  \end{tikzpicture}
\end{center}
```


Diferentes opciones para personalizar un modo como la separación, márgenes y bordes:

1. `/pgf/inner sep=<dimension> initial 0.3333em`. Esta opción especifica la separación interior de un nodo.

   Ejemplo 9.122

```
\begin{center}
  \begin{tikzpicture}
    \node[shape=circle, draw] at (0,0) {Heber MQ};
    \node[shape=circle, draw, inner sep=5mm] at (3,0)
    ↪ {Heber MQ};
  \end{tikzpicture}
\end{center}
```

2 `/pgf/inner xsep=<dimension> initial 0.3333em`. Esta opción especifica la separación interior en el eje x de un nodo.

Ejemplo 9.123

```
\begin{center}
  \begin{tikzpicture}
    \node[shape=rectangle, draw] at (0,0) {Heber MQ};
    \node[shape=rectangle, draw, inner xsep=5mm] at
    ↪  (3,0) {Heber MQ};
  \end{tikzpicture}
\end{center}
```

Heber MQ Heber MQ

3 `/pgf/inner ysep=<dimension> initial 0.3333em`. Esta opción especifica la separación interior en el eje y de un nodo.

Ejemplo 9.124

```
\begin{center}
  \begin{tikzpicture}
    \node[shape=rectangle, draw] at (0,0) {Heber MQ};
    \node[shape=rectangle, draw, inner ysep=5mm] at
    ↪  (3,0) {Heber MQ};
  \end{tikzpicture}
\end{center}
```

Heber MQ Heber MQ

4 `/pgf/outer sep=<dimension or "auto">`. Esta opción especifica la separación exterior de un nodo.

Ejemplo 9.125

```
\begin{center}
  \begin{tikzpicture}[every
   ↪  node/.style={fill=Salmon!50!white}]
    \node (a) at (0,0) {A};
    \node (b) at (4,0) {B};
    \node[outer sep=5mm] (c) at (0,-2) {C};
    \node[outer sep=5mm] (d) at (4,-2) {D};
    \draw (a) -- (b);
    \draw (c) -- (d);
  \end{tikzpicture}
\end{center}
```

A B

C D

5 `/pgf/outer xsep=<dimension>`. Esta opción especifica la separación exterior en el eje x de un nodo.

6 `/pgf/outer ysep=<dimension>`. Esta opción especifica la separación exterior en el eje y de un nodo.

7 `/pgf/minimum height=<dimension>`. Esta opción especifica la altura de un nodo.

Ejemplo 9.126

```
\begin{center}
  \begin{tikzpicture}
    \node[shape=rectangle, draw] at (0,0) {Heber MQ};
    \node[shape=rectangle, draw, minimum height=2cm] at
    ↪  (3,0) {Heber MQ};
  \end{tikzpicture}
\end{center}
```

Heber MQ Heber MQ

8 `/pgf/minimum width=<dimension>`. Esta opción especifica el ancho de un nodo.

Ejemplo 9.127

```
\begin{center}
  \begin{tikzpicture}
    \node[shape=rectangle, draw] at (0,0) {Heber MQ};
    \node[shape=rectangle, draw, minimum width=3cm] at
    ↪  (3,0) {Heber MQ};
  \end{tikzpicture}
\end{center}
```

Heber MQ Heber MQ

9 `/pgf/minimum size=<dimension>`. Esta opción especifica las dimensiones de altura y ancho del nodo.

Ejemplo 9.128

```
\begin{center}
  \begin{tikzpicture}
    \node[shape=rectangle, draw] at (0,0) {Heber MQ};
    \node[shape=rectangle, draw, minimum size=2cm] at
    ↪  (3,0) {Heber MQ};
  \end{tikzpicture}
\end{center}
```

Heber MQ Heber MQ

Para partir nodos debemos de utilizar el comando

```
\nodepart[<options>]{<part name>}
```

Este que es solamente funciona dentro del argumento obligatorio del comando `node`.

Ejemplo 9.129

```
\begin{center}
  \begin{tikzpicture}
    \node[fill=Salmon!70!white,shape=circle split]
    {
      $a$
      \nodepart{lower}
      $b$
    };
  \end{tikzpicture}
\end{center}
```

a

b

Color de texto de un nodo:

```
/tikz/text=<color>
```

Ejemplo 9.130

```
\begin{center}
    \tikz\node[text=red]{Heber MQ};
\end{center}
```

Heber MQ

Tipos de fuente de texto en un nodo:

```
/tikz/node font=<font commands>
/tikz/font=<font commands>
```

Ejemplo 9.131

```
\begin{center}
    \tikz\node[node font=\itshape]{Heber MQ};
    \tikz\node[font=\small]{Heber MQ};
\end{center}
```

Heber MQ Heber MQ

Formas para alinear un texto:

Ejemplo 9.132

```
\begin{center}
  \begin{tikzpicture}
    \node[draw] {
      \begin{tabular}{cc}
        upper left & upper right \\
        lower left & lower right
      \end{tabular}
    };
  \end{tikzpicture}
\end{center}
```

upper left	upper right
lower left	lower right

Para saltar a la siguiente línea, también podemos utilizar, \\, pero es necesario que asignemos una alineación.

Ejemplo 9.133

```
\tikz\node[draw,align=left]{Primera línea \\ Segunda
↪ línea};
```

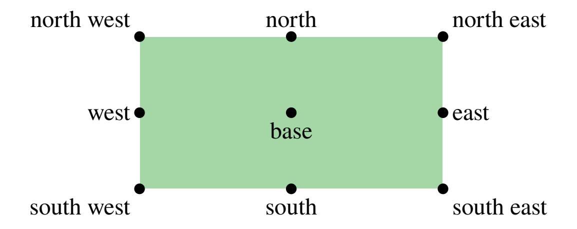
Primera línea
Segunda línea

Ejemplo 9.134

```
\tikz\node[draw,align=right]{Primera línea \\ Segunda
↪ línea};
```

Primera línea
Segunda línea

Anclas de un nodo:



Código de esta figura:

```
\begin{tikzpicture}
  \node[minimum width=4cm, minimum height=2cm, fill=g2, name=box]{};
  \node[above left] at (box.north west) {north west};
  \node[above] at (box.north) {north};
  \node[above right] at (box.north east) {north east};
  \node[left] at (box.west) {west};
  \node[right] at (box.east) {east};
  \node[below] at (box.base) {base};
  \node[below left] at (box.south west) {south west};
  \node[below right] at (box.south east) {south east};
```

```
    \node[below] at (box.south) {south};
    %
    \foreach \position in {north west, north, north east, west, base, east,
    ↪  south west, south, south east}
      \fill (box.\position) circle (2pt);
  \end{tikzpicture}
```

Cuando creamos un nodo, e indicamos que se ubique en las coordenadas `(0,0)`, centro del nodo se ubica en la coordenada `(0,0)`.

Ejemplo 9.135

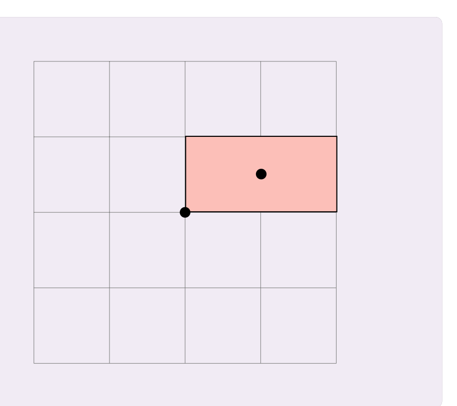
```
\begin{center}
  \begin{tikzpicture}
    \draw[help lines] (-2,-2) grid (2,2);
    \node[draw, minimum width = 2cm, minimum height = 1cm,
    ↪  fill=Salmon!50!white] (caja) at (0,0) {};
    \fill (caja.center) circle (2pt);
  \end{tikzpicture}
\end{center}
```

Podemos indicar a `tikz` que el anclaje sea una de las esquinas del nodo, y éste se logra con el key

```
/tikz/anchor=<anchor name>
```

Ejemplo 9.136

```
\begin{center}
  \begin{tikzpicture}
    \draw[help lines] (-2,-2) grid (2,2);
    \node[draw, anchor=south west, minimum width = 2cm,
    ↪  minimum height = 1cm, fill=Salmon!50!white] (caja)
    ↪  at (0,0) {};
    \fill (caja.center) circle (2pt);
    \fill (caja.south west) circle (2pt);
  \end{tikzpicture}
\end{center}
```



Ejemplo 9.137

```
\begin{center}
  \begin{tikzpicture}[scale=3,transform shape]
    \draw[anchor=center] (0,1)  node{x} -- (0.5,1)
    ↪ node{y} -- (1,1)  node{t};
    \draw[anchor=base]   (0,.5) node{x} -- (0.5,.5)
    ↪ node{y} -- (1,.5) node{t};
    \draw[anchor=mid]    (0,0)  node{x} -- (0.5,0)
    ↪ node{y} -- (1,0)  node{t};
  \end{tikzpicture}
\end{center}
```

x y t

x y t

x y t

Posicionamiento de un nodo con respecto a una coordenada:

```
/tikz/above=<offset>
/tikz/below=<offset>
/tikz/left=<offset>
/tikz/right=<offset>
/tikz/above left=<offset>
/tikz/above right=<offset>
/tikz/below left=<offset>
/tikz/below right=<offset>
```

Ejemplo 9.138

```
\begin{center}
  \begin{tikzpicture}
    \fill (0,0) circle (2pt) node[above] {texto por encima
    ↪ del punto};
  \end{tikzpicture}
\end{center}
```

texto por encima del punto

Para acceder a las opciones avanzadas de posicionamiento de un nodo se debe utilizar la librería `\usetikzlibrary{positioning}`. Posicionamiento de un nodo por encima de una coordenada determinada,

```
/tikz/above=<specification>
```

Ejemplo 9.139

```
\begin{center}
  \begin{tikzpicture}
    \fill (0,0) circle (2pt) node[above=5mm+2mm] {texto
    ↪ por encima del punto};
  \end{tikzpicture}
\end{center}
```

texto por encima del punto

Para posicionar un nodo con respecto a las anclas de otro nodo es necesario poner un nombre al nodo referencia, y así utilizar sus anclas:

Ejemplo 9.140

```
\begin{center}
  \begin{tikzpicture}
    \node[draw, minimum width = 2cm, minimum height = 1cm,
    ↪  fill=Salmon!50!white] (caja) at (0,0) {};
    \node[above=1cm of caja.north west, draw] {Nodo};
  \end{tikzpicture}
\end{center}
```

Nodo

En vez del key `above` se puede utilizar `below`, `left`, `right`, `above left`, `above right`, `below left` y `below right`. Siempre se debe utilizar `of` para hacer deferencia al ancla de otro modo.

Ejemplo 9.141

```
\begin{center}
  \begin{tikzpicture}
    \node[draw, minimum width = 2cm, minimum height = 1cm,
    ↪  fill=Salmon!50!white] (caja) at (0,0) {};
    \node[above=2cm of caja, draw] {Nodo};
  \end{tikzpicture}
\end{center}
```

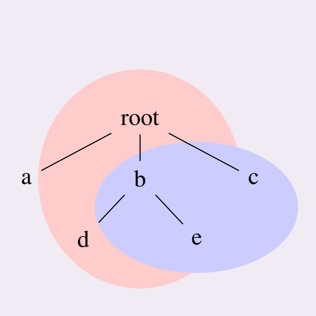
Nodo

Tamaño de un nodo dado por un conjunto de puntos:

Ejemplo 9.142

```
\begin{center}
  \begin{tikzpicture}[level distance=8mm]
    \node (root) {root}
      child { node (a) {a} }
      child { node (b) {b}
        child { node (d) {d} }
        child { node (e) {e} } }
      child { node (c) {c} };

    \begin{scope}[on background layer]
      \node[fill=red!20,inner sep=0pt,ellipse,fit=(root)
      ↪  (b) (d) (e)] {};
      \node[fill=blue!20,inner sep=0pt,ellipse,fit=(b) (c)
      ↪  (e)] {};
    \end{scope}
  \end{tikzpicture}
\end{center}
```



Nodos a lo largo una curva:

1. `/tikz/pos=<fraction>`. Este key lo ubica al nodo a distancia (en tanto por uno).

Ejemplo 9.143

```
\begin{center}
  \begin{tikzpicture}
    \draw(0,0) -- (4,1) node[pos=0]{0}
    ↪  node[pos=0.5]{0.5} node[pos=0.85]{0.85}
    ↪  node[pos=1]{1};
  \end{tikzpicture}
\end{center}
```


(a) El valor `pos=0` es equivalente al key `/tikz/at start`

(b) El valor `pos=0.25` es equivalente al key `/tikz/near start`.

(c) El valor `pos=0.5` es equivalente al key `/tikz/midway`.

(d) El valor `pos=0.75` es equivalente al key `/tikz/near end`.

(e) El valor `pos=1` es equivalente al key `/tikz/at end`.

2. `/tikz/sloped`. Este key hace girar al nodo de tal modo que su línea base sea paralelo a la línea o a la recta tangente de una curva.

Ejemplo 9.144

```
\begin{center}
  \begin{tikzpicture}
    \draw[-latex] (0,0) -- (2,2) node[above, midway,
    ↪  sloped] {$\vec{x}$};
  \end{tikzpicture}
\end{center}
```


Graph

Para crear un grafo se debe utilizar la librería `\usetikzlibrary{graphs}`.

Ejemplo 9.145

```
\tikz\graph { a -> {b, c} -> d };
```


Para más información acerca de este libro, se recomienda al lector revisar la documentación oficial de `tikz`.

Transparency

Para establecer la opacidad de un trazado se debe utilizar el key:

```
/tikz/draw opacity = <value>
```

Los valores de <value> varían de 0 a 1.

Ejemplo 9.146

```
\begin{center}
  \begin{tikzpicture}
    \draw[line width=5pt] (0,0) -- (4,0);
    \draw[line width=5pt, draw opacity=0.5] (0,-1) --
    ↪  (4,-1);
  \end{tikzpicture}
\end{center}
```

Para establecer la opacidad de un relleno se debe utilizar el key:

```
/tikz/fill opacity = <value>
```

Los valores de <value> varían de 0 a 1.

Ejemplo 9.147

```
\begin{tikzpicture}[thick,fill opacity=0.5]
  \filldraw[fill=red]   (0:1cm)    circle (12mm);
  \filldraw[fill=green] (120:1cm)  circle (12mm);
  \filldraw[fill=blue]  (-120:1cm) circle (12mm);
\end{tikzpicture}
```

Para establecer la transparencia de un grupo de dibujos se debe de utilizar el key:

```
/tikz/opacity=<value>
```

Ejemplo 9.148

```
\begin{center}
  \begin{tikzpicture}
    \draw[fill=Salmon!50!white, line width=2pt] (0,0)
    ↪ rectangle (2,1);
    \draw[fill=Salmon!50!white, line width=2pt,
    ↪ opacity=0.5] (2.5,0) rectangle (4.5,1);
  \end{tikzpicture}
\end{center}
```

Decorate paths

Para decorar un path es necesario importar la librería `decorations`. La sintaxis para decorar un path es:

```
\path ... decorate[<options>]{<subpath>} ...;
```

Ejemplo 9.149

```
\tikz\draw decorate [decoration={name=zigzag}] {(0,0) --
↪ (4,0)};
```

También podemos lograr lo mismo del siguiente modo:

Ejemplo 9.150

```
\begin{center}
  \begin{tikzpicture}
    \draw                                                   (0,3)   -- (3,3);
    \draw[decorate,decoration=zigzag]                       (0,2.5) -- (3,2.5);
    \draw[decorate,decoration=brace]                        (0,2)   -- (3,2);
    \draw[decorate,decoration=triangles]                    (0,1.5) -- (3,1.5);
    \draw[decorate,decoration={coil,segment length=4pt}] (0,1)   -- (3,1);
    \draw[decorate,decoration={coil,aspect=0}]              (0,.5)  -- (3,.5);
    \draw[decorate,decoration={expanding waves,angle=7}] (0,0)   -- (3,0);
  \end{tikzpicture}
\end{center}
```

El key `/pgf/decoration=<decoration options>` es utilizado para establecer el decoración que se va a utilizar, y además, es utilizado para indicar las diferentes opciones de la decoración.

Ejemplo 9.151

```
\tikz\draw decorate [decoration=zigzag] {(0,0) -- (4,0)};
```

El key

```
/pgf/decoration/name=<name>
```

es utilizado para establecer el nombre de la decoración a utilizar (está disponible solo en tikz).

Librerias

Lista de librerias

1. Angle.
2. Arrow.
3. Babel.
4. Background.
5. Calc.
6. Calendar.
7. Chains.
8. Circuit.
9. Decoration.
10. er
11. External.
12. Fading.
13. Fit.
14. Fixedpointarithmetic.
15. Fpu.
16. Lindenmayersystems.
17. Math.
18. Matrix.
19. Mindmap.
20. Folding.
21. Pattern.
22. Petri-Net.
23. Plothandlers.
24. Plotmarks.
25. Profiler.
26. Shadings.
27. Shadows.
28. Shape.
29. Spy.
30. Svg.path.
31. Topath.
32. Through.
33. Tree.
34. Turtle.

Utilidades

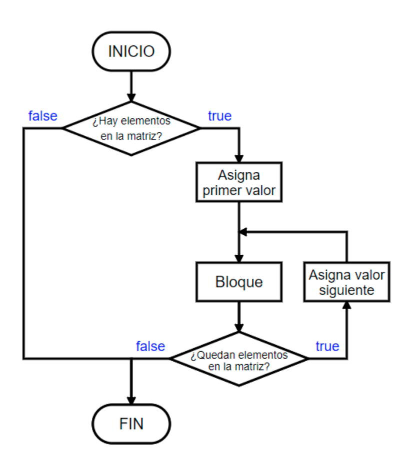
Bucle foreach

Hasta el momento hemos utilizado el comando `\foreach`, este comando es proporcionado por el paquete `pgffor`.

Sintaxis del comando `\foreach`:

```
\foreach<variables>[<options>]in<list> <commands>
```

Figura 9.2



Ejemplo 9.152

```
\foreach \x in {1,2,3,0} {[\x]}
```

[1][2][3][0]

Ejemplo 9.153

```
\def\mylist{1,2,3,0}
\foreach \x in \mylist {[\x]}
```

[1][2][3][0]

Combinando `\foreach` con `tikz`:

Ejemplo 9.154

```
\begin{center}
  \begin{tikzpicture}
    \foreach \x in {0,1,2,3}
      \draw (\x,0) circle (0.2cm);
  \end{tikzpicture}
\end{center}
```

```
\begin{center}
  \begin{tikzpicture}
    \foreach \x in {0,1,2,3}
      \foreach \y in {0,1,2,3}
      {
        \draw (\x,\y) circle (0.2cm);
        \fill (\x,\y) circle (0.1cm);
      }
  \end{tikzpicture}
\end{center}
```

Motor matemático

Para realizar operaciones matemáticas es necesario utilizar el paquete pgfmath.

Expresiones matemáticas

El argumento del comando

```
\pgfmathparse{<expression>}
```

puede recibir cualquier expresión matemática y retorna el macro

```
\pgfmathresult
```

Operaciones más conocidas

Ejemplo 9.155

```
\pgfmathparse{2+2}\pgfmathresult\\
\pgfmathparse{3-2}\pgfmathresult\\
\pgfmathparse{3*2}\pgfmathresult\\
\pgfmathparse{6/2}\pgfmathresult\\
\pgfmathparse{6^2}\pgfmathresult\\
\pgfmathparse{4!}\pgfmathresult
```

4.0
1.0
6.0
3.0
36.0
24.0

Funciones

Tabla 9.20

abs	acos	add	and	array	asin
atan	atan2	bin	ceil	cos	cosec
cosh	cot	deg	depth	div	divide
e	equal	factorial	false	floor	frac
gcd	greater	height	hex	Hex	int
ifthenelse	iseven	isodd	isprime	less	ln
log10	log2	max	min	mod	Mod
multiply	neg	not	notequal	notgreater	notless
oct	or	pi	pow	rad	rand
random	rnd	round	scalar	sec	sign
sin	sinh	sqrt	subtract	tan	tanh
true	veclen	width			

Estes funciones se deben utilizar de la siguiente manera:

Ejemplo 9.156

```
\pgfmathparse{add(2,1)}\pgfmathresult\\
\pgfmathparse{sqrt(4)}\pgfmathresult\\
\pgfmathparse{(e^2-e^-2)/2}\pgfmathresult\\
\pgfmathparse{pi}\pgfmathresult
```

3.0
2.00000
3.62685
3.141592654

ASYMPTOTE

10.1 Primeros pasos en asymptote

Hola mundo

```
label("Hello world");
```

Se obtiene: Hello world

10.2 Gráficas en dos dimensiones

Lines, paths, clips, sizing and transformations

Size figure

En esta sección estudiaremos a definir el tamaño de las figuras generados por asymptote.

```
size(5,5) //tamaño rigido
size(0,5) //en x es expansible
size(5) //tamaño rigido - equivale a size(5,5)
```

Una unidad set del siguiente modo `unitsize(1cm)`.

Labels

Podemos generar etiquetas en Asymptote de manera similar a como lo hacemos con el paquete TikZ. Por ejemplo,

Ejemplo 10.1

```
size(3cm);
draw(unitsquare);
label("$A$",(0,0),SW);
label("$B$",(1,0),SE);
label("$C$",(1,1),NE);
label("$D$",(0,1),NW);
```

D C A B

Para crear una etiqueta en Asymptote, es necesario primero definir la posición en la que se quiere que aparezca la etiqueta. Por ejemplo, si se desea agregar una etiqueta en el punto (1,1), se puede realizar lo siguiente:

```
pair p = (1,1);
label("$P$", p);
```

En el código anterior, la variable p almacena la posición del punto (1,1) y la función label crea una etiqueta con el texto "*P*", la cual se ubica en la posición almacenada en p.

Es posible ajustar la posición y la apariencia de la etiqueta mediante los parámetros opcionales de la función label. Si se desea que la etiqueta esté más alejada del punto, se puede utilizar el parámetro align para especificar la alineación de la etiqueta con respecto a la posición. Por ejemplo, si se quiere que la etiqueta se encuentre a la derecha del punto, se puede realizar lo siguiente:

```
pair p = (1,1);
label("$P$", p, E);
```

En el código anterior, el parámetro E indica que la etiqueta se alinee a la derecha (E proviene de "east" en inglés).

La función label tiene la syntaxis:

```
void label(picture pic=currentpicture, Label L,
  pair position, align align=NoAlign,
  pen p=nullpen, filltype filltype=NoFill);
```

Paths

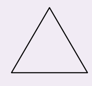
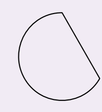
Formas básicas: `p--q`, `p..q`, `p^^q`. Se pueden utilizar las siguientes opciones para unir dos puntos en Asymptote:

1. La opción `p--q` une una línea recta desde el punto p hasta el punto q.
2. La opción `p..q` une una línea desde el punto p hasta el punto q con una interpolación spline cúbica de Bézier.

3 La opción p^^q no une una línea desde el punto p hasta el punto q, sino que hace que los paths se traten como si fueran uno solo.

Ejemplo 10.2

```
unitsize(1cm);
path T,ct,tt;
T=(0,0)--(1,0)--(1/2,sqrt(3)/2)--cycle;
ct=(0,0)..(1,0)--(1/2,sqrt(3)/2)..cycle;
tt=shift(sqrt(3)/6*dir(30))*(scale(1/2)*T);
draw(T);
draw(shift(2*right)*ct);
fill(reverse(shift(4*right)*tt)^^(shift(4*right)*T),blue);
```

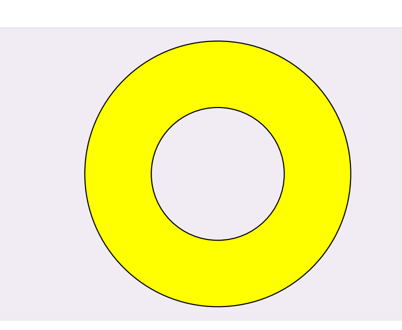




Ejemplo 10.3

```
unitsize(1cm);
size(5,0);
draw((1,0){up}..{left}(0,1),Arrow);
```

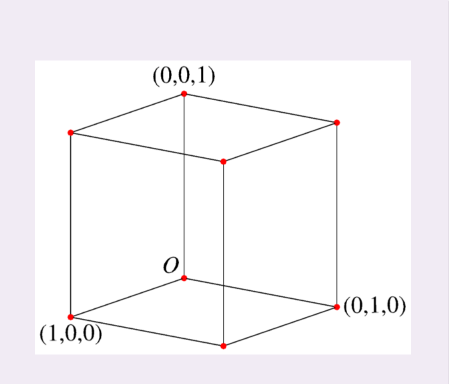
Ejemplo 10.4

```
size(0,100);
path unitcircle=E..N..W..S..cycle;
path g=scale(1.5)*unitcircle;
filldraw(unitcircle^^g,evenodd+yellow,black);
```



Ejemplo 10.5

```
import three;
currentprojection=orthographic(5,4,2,center=true);
size(5cm);
size3(3cm,5cm,8cm);
draw(unitbox);
dot(unitbox,red);
label("$O$",(0,0,0),NW);
label("(1,0,0)",(1,0,0),S);
label("(0,1,0)",(0,1,0),E);
label("(0,0,1)",(0,0,1),Z);
```



Pen

Un tipo de dato pen es nada mas que los colores y los estilos de línea.

Ejemplo 10.6

```
unitsize(1cm);
draw((0,0)--(2,0),orange+dashed+linewidth(1));
```

El sigo + es un operador de mixing pen.

Draw

La función draw tiene la syntaxis:

```
void draw(picture pic=currentpicture, Label L="", path g,
  align align=NoAlign, pen p=currentpen,
  arrowbar arrow=None, arrowbar bar=None, margin margin=NoMargin,
  Label legend="", marker marker=nomarker);
```

En donde path g es el argumente obligatorio.

Fill

La función fill tiene la syntaxis:

```
void fill(picture pic=currentpicture, path g,pen p=currentpen);
```

Ejemplo:

Ejemplo 10.7

```
unitsize(1cm);
fill((0,0)--(0,1)--(1,1)--(1,0)--cycle,red);
```

Otra estructura:

```
void filldraw(picture pic=currentpicture,
            path g, pen fillpen=currentpen,
            pen drawpen=currentpen);
```

Clip

La función clip tiene la syntaxis:

```
void clip(picture pic=currentpicture, path g, pen p=currentpen);
```

Recorta el imagen con límite path g y rellena con p.

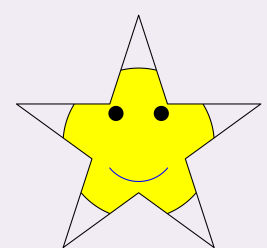
Regla de relleno: zerowinding, fillrule, evenodd y fillrule.

Ejemplo 10.8

```
import graph;
unitsize(1cm);
filldraw(Circle((0,0),1),yellow,black);
fill(Circle((-.3,.4),.1),black);
fill(Circle((.3,.4),.1),black);
draw(Arc((0,0),.5,-140,-40),blue);
```

Ejemplo 10.9

```
import graph;
unitsize(1cm);
filldraw(Circle((0,0),1),yellow,black);
fill(Circle((-.3,.4),.1),black);
fill(Circle((.3,.4),.1),black);
draw(Arc((0,0),.5,-140,-40),blue);
path star;
star=expi(0)--(scale((3-sqrt(5))/2)*expi(pi/5))--expi(2*pi/5)--
(scale((3-sqrt(5))/2)*expi(3*pi/5))--expi(4pi/5)--
(scale((3-sqrt(5))/2)*expi(5*pi/5))--expi(6*pi/5)--
(scale((3-sqrt(5))/2)*expi(7*pi/5))--expi(8*pi/5)--
(scale((3-sqrt(5))/2)*expi(9*pi/5))--cycle;
clip(currentpicture,scale(1.7)*rotate(18)*star);
draw(scale(1.7)*rotate(18)*star);
```



Pictures

Comandos básicos: draw, fill, clip y path, dibujan en currentpicture.

Transformaciones

Las transformaciones son objetos que el operador $*$ puede aplicar a pares, trazados, imágenes y otras transformaciones (es como transformaciones en geometría analítica). Las transformaciones más utilizadas son:

```
shift(pair p)
rotate(real angle, pair p = (0,0))
xscale(real r)
yscale(real r)
scale(real r)
```

Otro función de transformación es: `reflect(pair p, pair q) //line p--q`

Arrowheads

Algunos tipos de flechas se muestra en la tabla 10.21.

Ejemplo 10.10

```
unitsize(1cm);
draw((-.1,0) -- (1,0), arrow=Arrow);
draw((0,-.1) -- (0,1), arrow=Arrow);
```

Código	Resultado
`draw((0,0)--(1,0),arrow=Arrow());`	
`draw((0,0)--(1,0),arrow=ArcArrow());`	
`draw((0,0)--(1,0),arrow=Arrow(SimpleHead));`	
`draw((0,0)--(1,0),arrow=ArcArrow(SimpleHead));`	
`draw((0,0)--(1,0),arrow=Arrow(HookHead));`	
`draw((0,0)--(1,0),arrow=ArcArrow(HookHead));`	
`draw((0,0)--(1,0),arrow=Arrow(TeXHead));`	

Tabla 10.21 Diferentes tipos de flechas

Curved paths

Ejemplo 10.11

```
unitsize(1cm);
draw((0,0) .. (1,1) .. (2,sqrt(2)));
```

Otra forma:

Ejemplo 10.12

```
unitsize(1cm);
draw((0,0) .. controls (1,1) and (2,1) .. (3,0));
```

Markers on paths

La opción "marker" permite mostrar los puntos de un camino como pequeños marcadores en la imagen generada. Por ejemplo, si se tiene un camino definido por varios puntos, utilizando la opción "marker" se pueden mostrar esos puntos en la imagen resultante, lo que puede ser útil para verificar que el camino se ha definido correctamente.

Ejemplo 10.13

```
settings.outformat="pdf";
unitsize(0.5cm);
draw((0,0) .. (pi/2,1) .. (pi,0) .. (3*pi/2,-1) .. (2*pi,
↪  0),
marker=MarkFill[0]);
```

En vez de `MarkFill` se puede utilizar `Mark`. Valores de índice para `Mark` y `MarkFill` son:

1. `[0]`: circulo.
2. `[1]`: triángulo.
3. `[2]`: cuadrado.
4. `[3]`: pentágono.
5. `[4]`: triángulo invertido.
6. `[5]`: marcas x.
7. `[6]`: asterisco.

Circles and ellipses

La función circle tiene la syntaxis: `path circle(pair c,real r);`. En donde c es el centro del círculo y r es el radio.

La función ellipse tiena la syntaxis: `path ellipse(pair c,real a,real b);`. En donde c es el centro del círculo, a radio horizontal y b radio vertical.

Ejemplo 10.14

```
size(2cm);
draw(circle((0,1), 0.5), red);
draw(circle((1,0), 1.5), blue);
draw(ellipse((1,0), 1.5, 0.5));
```

Boxes and polygons

La función box tiena la syntaxis: `path box(pair a,pair b);`. En donde a es el extremo inferior izquierdo y b es el extremo superior derecho.

Ejemplo 10.15

```
size(1cm);
draw(box((0,0),(1,1)), red);
```

La función polygon tiena la syntaxis: `path polygon(int n);`. En donde n es el número de lados.

Ejemplo 10.16

```
size(1cm);
draw(unitcircle);
draw(polygon(5), blue);
```

Transformations: shifting, scaling, rotating, etc.

Ejemplo 10.17

```
size(5cm);
for (int n = 3; n <= 7; ++n) {
  draw(shift(2.2*n, 0) *
  polygon(n));
}
```

La estructura de control for es similar al lenguaje C. En lenguja C se utiliza n++ para incrementar n en una unidad. En asympote se utiliza ++n para incrementar n en una unidad, debido a que se puede confundir con $n--$,

Ejemplo 10.18

```
size(3cm,0);
path p = box((0,0), (1,1));
draw(p, black + linewidth(2.0pt));
draw(shift(1,2)*p, blue);
draw(xscale(1.6)*p, green);
draw(yscale(1.4)*p, orange);
draw(scale(1.8)*p, red);
draw(rotate(60)*p, purple); /*Rotate 60
degrees*/
```

Arcs and margins

Función arc tiena la syntaxis: `path arc(pair c, real r, real angle1, real angle2);`. En donde c es el centro del arco, r es el radio, $angle1$ es el ángulo inicial y $angle2$ es el ángulo final en grados.

Ejemplo 10.19

```
size(2cm,0)
draw((3,0)--(0,0)--(0,4));
draw((2,1) -- arc((2,1), 2, 60, 80) -- cycle);
```

Ejemplo 10.20

```
size(3cm,0);
draw((-1.2,0)--(1.2,0));
draw((0,-1.2)--(0,1.2));
/* An arc from 270 to 0 goes clockwise. */
draw(arc((0,0), r=1, angle1=270, angle2=0),
arrow=Arrow(TeXHead));
```

Otra función arc tiena la syntaxis:

```
path arc(pair c, explicit pair z1,
        explicit pair z2,
        bool direction = CCW);
```

En donde c es el centro del arco, $z1$ es el punto inicial, $z2$ es el punto final y CCW (counterclockwise) es una variable booleana que indica si el arco es en sentido horario o antihorario.

Ejemplo 10.21

```
size(3cm,0);
draw((3,0) -- (0,0) -- (3,4));
draw(arc((0,0), (2,0), (3,4)),
arrow=Arrow(TeXHead), red);
draw(arc((0,0), (2,0), (3,4), direction=CW),
arrow=Arrow(TeXHead), blue);
dot((0,0));
dot((2,0));
dot((3,4));
```

Filling a region

Para sombrear una región debajo de una curva se debe de utilizar la función fill.

Ejemplo 10.22

```
settings.outformat="pdf";
unitsize(1cm);
fill((0,0){up} .. (1,1) .. (2,sqrt(2)) -- (2,0) -- cycle,
↪ mediumgray);
draw((-.1,0) -- (2,0), arrow=Arrow(TeXHead));
draw((0,-.1) -- (0,2), arrow = Arrow(TeXHead));
draw((0,0){up} .. (1,1) .. (2,sqrt(2)));
```

Drawing a dot at a point

Ejemplo 10.23

```
size(2.5cm,2.5cm, keepAspect=false);
draw((0,0) -- (50,0), arrow=Arrow(TeXHead));
draw((0,0) -- (0,10),
arrow=Arrow(TeXHead));
dot((2,1));
dot((35,8));
dot((42,9), red);
```

Named paths and variables

Para declarar a un path o nombrar a un path se debe recurir al tipo de dato path.

Ejemplo 10.24

```
unitsize(0.5cm);
path s = (0,0){up} .. (1,1) .. (2,sqrt(2));
draw(s, red);
```

Path times and subpaths

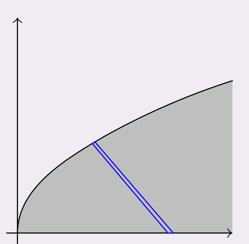
La sintaxis de la función subpaths es: `path subpath(path p, real a, real b);`. En donde p es el path original y a y b son los índices de los subpaths que queremos obtener. Esta función obtiene un trozo de path del path p (original) desde el longitud de arco a (time a) hasta la longitud de arco b (time b). Si $a > b$, se invierte la dirección de subpath.

Ejemplo 10.25

```
size(2cm);
real xmin = -0.1;
real xmax = 2;
real ymin = -0.1;
real ymax = 2;
path s = (0,0){up} .. (1,1) .. (xmax,sqrt(xmax));
fill(s -- (xmax,0) -- cycle, mediumgray);
draw(s);
arrowbar axisarrow = Arrow(TeXHead);
draw((xmin,0) -- (xmax,0), arrow=axisarrow);
draw((0,ymin) -- (0,ymax), arrow = axisarrow);
real x = 1.4;
real dx = .05;
path striptop = subpath(s,x,x+dx);
draw((x,0) -- striptop -- (x+dx,0) -- cycle, blue);
clip(box((1,ymin),(1.6,1.5)));
```

Ejemplo 10.26

```
size(3cm,0);
real xmin = -0.1;
real xmax = 2;
real ymin = -0.1;
real ymax = 2;
path s = (0,0){up} ..
(1/2,sqrt(1/2)) .. (1,1) .. (xmax,sqrt(xmax));
fill(s -- (xmax,0) -- cycle, mediumgray);
draw(s);
arrowbar axisarrow = Arrow(TeXHead);
draw((xmin,0) -- (xmax,0), arrow=axisarrow);
draw((0,ymin) -- (0,ymax), arrow = axisarrow);
real x = 1.4;
real dx = .05;
path striptop = subpath(s,x,x+dx);
draw((x,0) -- striptop -- (x+dx,0) -- cycle, blue);
```



Intersections and arrays and subpaths

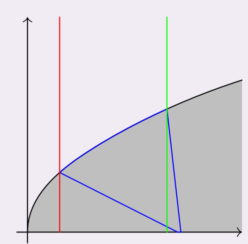
La función `real[] times(path p, real x);` devuelve un array con los tiempos de los arcos de p que cruzan el eje x en el punto x. El tiempo de un arco es la longitud del arco desde el punto de partida hasta el punto de cruce.

Ejemplo 10.27

```
size(0,3cm);
real xmin = -0.1;
real xmax = 2;
real ymin = -0.1;
real ymax = 2;
path s = (0,0){up} .. (1/2,sqrt(1/2)) .. (1,1) .. (xmax,sqrt(xmax));
fill(s -- (xmax,0) -- cycle, mediumgray);
draw(s);
arrowbar axisarrow = Arrow(TeXHead);
draw((xmin,0) -- (xmax,0), arrow=axisarrow);
draw((0,ymin) -- (0,ymax), arrow = axisarrow);
real x = 1.4;
real dx = .03;
real x1 = 0.3;
real x2 = x1+1;
real t0 = times(s,x1)[0];
real t1 = times(s,x2)[0];
path striptop = subpath(s,t0,t1);
draw((x1,0) -- striptop -- (x2,0) -- cycle, blue);
clip(box((1,ymin),(1.6,1.5)));
```

Ejemplo 10.28

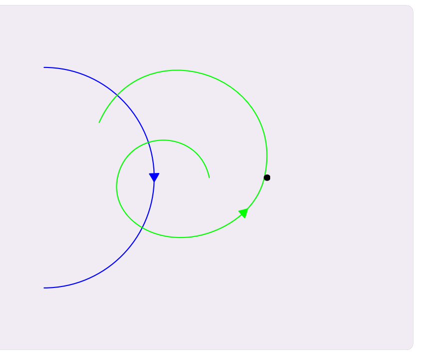
```
size(3cm,0);
real xmin = -0.1;
real xmax = 2;
real ymin = -0.1;
real ymax = 2;
path s = (0,0){up} .. (1/2,sqrt(1/2)) .. (1,1) .. (xmax,sqrt(xmax));
fill(s -- (xmax,0) -- cycle, mediumgray);
draw(s);
arrowbar axisarrow = Arrow(TeXHead);
draw((xmin,0) -- (xmax,0), arrow=axisarrow);
draw((0,ymin) -- (0,ymax), arrow = axisarrow);
real x = 1.4;
real dx = .03;
real x1 = 0.3;
real x2 = x1+1;
real t0 = times(s,x1)[0];
real t1 = times(s,x2)[0];
path striptop = subpath(s,t0,t1);
draw((x,0) -- striptop -- (x+dx,0) -- cycle, blue);
draw((x1,0) -- (x1,2), red);
draw((x2,0) -- (x2,2), green);
```



Para determinar la intersección de dos paths, se puede usar la función `real[] intersect(path p, path q);` asumiendo que p y q se intersectan. La función devuelve un array con dos elementos, que corresponden a los puntos de cruce de p y q.

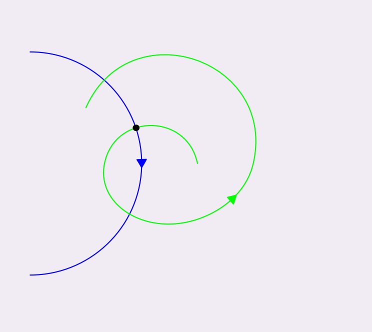
Ejemplo 10.29

```
size(3cm,0);
path p = (-1,1) .. (0,0) .. (-1,-1);
path q = (1/2,0) .. (-1/3,0) .. (1/2,-1/2) ..
  (1,0) .. (-1/2,1/2);
draw(p,blue, arrow=MidArcArrow());
draw(q,green, arrow=MidArcArrow());
real[] isections = intersect(p,q);
dot(point(isections[0]));
```



Ejemplo 10.30

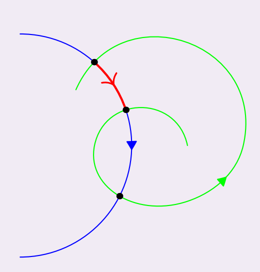
```
size(3cm,0);
path p = (-1,1) .. (0,0) .. (-1,-1);
path q = (1/2,0) .. (-1/3,0) .. (1/2,-1/2) ..
  (1,0) .. (-1/2,1/2);
draw(p,blue, arrow=MidArcArrow());
draw(q,green, arrow=MidArcArrow());
real[] isections = intersect(p,q);
dot(point(q,isections[1]));
```



En los ejemplos anteriores ilustrados, los respectivos paths tienen varios puntos de intersección y para obtener los puntos de intersección se utiliza la función intersections. La función `real[][] intersections(path p, path q);` devuelve un array de arrays, donde cada subarray contiene los puntos de cruce de p y q.

Ejemplo 10.31

```
size(3cm,0);
path p = (-1,1) .. (0,0) .. (-1,-1);
path q = (1/2,0) .. (-1/3,0) .. (1/2,-1/2) ..
↪ (1,0) .. (-1/2,1/2);
draw(p,blue, arrow=MidArcArrow());
draw(q,green, arrow=MidArcArrow());
real[][] a = intersections(p,q);
draw(subpath(p,a[0][0], a[1][0]),
↪ red+linewidth(0.8),
arrow=MidArrow(TeXHead));
//código adicional
dot(point(p,a[0][0]));
dot(point(p,a[1][0]));
dot(point(p,a[2][0]));
```



Sea los puntos m, n y t tiempos de intersección de p y x, y y z los tiempos de intersección de q, entonces, el array de arrays es de la forma como se puede ver en la tabla 10.22.

m	x
n	y
t	z

Tabla 10.22

Tambien se puede obtener el punto de intersección de dos paths, utilizando la función intersectionpoint.

```
pair intersectionpoint(path p, path q);
```

Esta función devuelve un par con los puntos de cruce de p y q.

La función,

```
pair[] intersectionpoint(path p, path q);
```

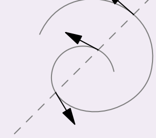
devuelve un array de pares, donde cada par contiene los puntos de cruce de p y q.

Tangent lines

La función `dir(path p, real t);` retorna la dirección del vector unitario tangente de path p en el tiempo t.

Ejemplo 10.32

```
size(2cm, 0);
path p = (1/2,0) .. (-1/3,0) .. (1/2,-1/2) .. (1,0) .. (-1/2,1/2);
path l = (-1,-1) -- (1,1);
draw(l,dashed+gray);
draw(p, gray);
for (real[] pathtime : intersections(p,l)) {
  real t = pathtime[0];
  pair tangent = dir(p, t);
  draw(shift(point(p,t)) * scale(1/2) * ((0,0) -- tangent),arrow=Arrow);
}
```



Drawing disconnected paths

Para descontinuar un path utilizar `^^`.

Ejemplo 10.33

```
size(5cm,0);
draw((0,0) -- (0,1) ^^ (1,0) .. (3/2,1) .. (2,0) ^^ (3,0) -- (4,1));
```

Graphing functions

Para graficar una función como $f(x) = \sqrt{x}$ se debe recurir a la librería graph.

Ejemplo 10.34

```
size(3cm);
import graph;
real f(real x){
  return sqrt(x);
}
path g = graph(f,0,3);//x in [0,3]
draw(g,blue);
```

Se puede utilizar el argumento operator .. o operator – para la gráfica de función.

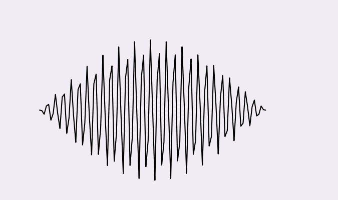
Ejemplo 10.35

```
size(3cm);
import graph;
real f(real x){
  return sqrt(x);
}
path g = graph(f,0,3,n=200, operator ..);//x in [0,3]
draw(g,blue);
```

Diferentes gráficas (compare):

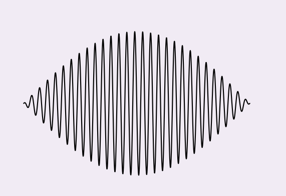
Ejemplo 10.36

```
import graph;
size(3cm);
real f(real x) { return sin(x)*cos(57*x); }
path g = graph(f,0, pi);
draw(g);
```



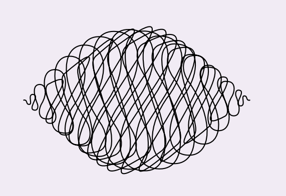
Ejemplo 10.37

```
import graph;
size(3cm);
real f(real x) { return sin(x)*cos(57*x); }
path g = graph(f,0, pi, n=1000);
draw(g);
```



Ejemplo 10.38

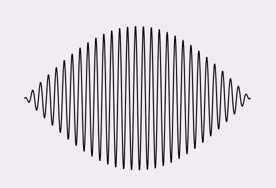
```
import graph;
size(3cm);
real f(real x) { return sin(x)*cos(57*x); }
path g = graph(f,0, pi, operator ..);
draw(g);
```



Otra de las opciones es Hermite.

Ejemplo 10.39

```
import graph;
size(3cm);
real f(real x) { return sin(x)*cos(57*x); }
path g = graph(f,0, pi, n=200, Hermite);
draw(g);
```



Parametric graphs

Sean

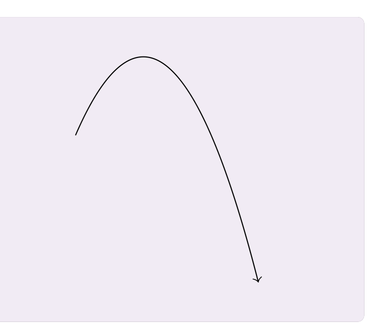
$$y(t) = -0.5t^2 + 3.0t + 1.0$$

$$x(t) = 1.3t$$

las las funciones paramétricas.

Ejemplo 10.40

```
size(3cm);
import graph;
pair F(real t) {
  return (1.3*t,-4.5*t^2+3.0*t+1.0);
}
path g = graph(F, 0, 0.9);
draw(g, arrow=Arrow(TeXHead));
```



Gráficas definidas implícitamente

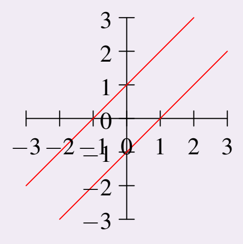
Asymptote tiene la capacidad de graficar curvas definidas implícitamente. Esto significa que en lugar de tener una función explícita de la forma y = f(x), se tiene una ecuación que relaciona x y y, pero no se puede despejar una variable en términos de la otra. Para graficar estas curvas, se utiliza el módulo “contour” en lugar del módulo “graph” de Asymptote. La función relevante es

```
contour(real f(real, real), pair a, pair b, real[] c);
```

Esta función devuelve `guide[][]` una matriz de matrices de guías. Si se guarda esta `guide[][]` como una variable llamada "thegraphs", entonces para cada i, la entrada "`thegraphs[i]`" es una `guide[]` que representa el gráfico de la ecuación "$f(x,y)=c[i]$".

Ejemplo 10.41

```
import graph;
import contour;
size(5cm,3cm);
real f(real x, real y) {
  return x^2 - 2*x*y + y^2 - 1;
}
pair a = (-3,-3);
pair b = (3,3);
real[] c = {0};
guide[][] thegraphs = contour(f, a, b, c);
draw(thegraphs[0], red);
xaxis(-3, 3, Ticks(Step=1.0, Size=1mm));
yaxis(-3, 3, Ticks(Step=1.0, Size=1mm));
```



En este ejemplo, se define una función “f” que toma dos argumentos (x y y) y devuelve el resultado de evaluar la ecuación $x^2-2xy+y^2-1$ en esos valores. Luego se definen las esquinas de la región rectangular en la que se desea graficar la curva (a y b) y el arreglo de valores c que contiene el nivel de contorno en el que se desea graficar la curva (en este caso, $c=\{0\}$ ya que se desea graficar la curva en el nivel de contorno 0).

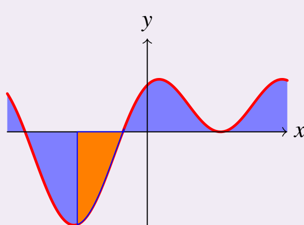
La función “contour” se utiliza para obtener la `guide[][]` de la curva, la cual se guarda en la variable “thegraphs”. Luego se utiliza la función “draw” para graficar la guía en el primer nivel de contorno (`thegraphs[0]`) en color rojo. Finalmente, se agregan los ejes x y y con etiquetas y marcas de graduación.

The filldraw command

El comando `filldraw` tiene la misma sintaxis que el comando `draw`, pero en lugar de sólo contornear un objeto, lo rellena con un color y también lo contornea si se desea.

Ejemplo 10.42

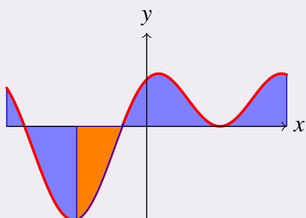
```
settings.outformat="pdf";
size(4cm,0);
import graph;
real xmin = -3;
real xmax = 3;
real ymin = -2;
real ymax = 2;
real f(real x) {
  return sin(x) + cos(2*x);
}
path s = graph(f, xmin, xmax, operator ..);
fill(s -- (xmax,0) -- (xmin,0) -- cycle, lightblue);
draw(s, linewidth(1bp) + red);
arrowbar axisarrow = Arrow(TeXHead);
draw((xmin,0) -- (xmax,0), arrow=axisarrow);
draw((0,ymin) -- (0,ymax), arrow=axisarrow);
label("$x$",(xmax,0),E);
label("$y$",(0,ymax),N);
real x = -1.5;
real dx = 1;
real t0 = times(s,x)[0];
real t1 = times(s,x+dx)[0];
path striptop = subpath(s,t0,t1);
fill(striptop -- (x+dx,0) -- (x,0) -- cycle, orange);
draw((x,0) -- striptop -- (x+dx,0) -- cycle, blue);
```



Utilizando comando filldraw

Ejemplo 10.43

```
settings.outformat="pdf";
size(4cm,0);
import graph;
real xmin = -3;
real xmax = 3;
real ymin = -2;
real ymax = 2;
real f(real x) {
  return sin(x) + cos(2*x);
}
path s = graph(f, xmin, xmax, operator ..);
filldraw(s -- (xmax,0) -- (xmin,0) -- cycle, lightblue, blue);
draw(s, linewidth(1bp) + red);
arrowbar axisarrow = Arrow(TeXHead);
draw((xmin,0) -- (xmax,0), arrow=axisarrow);
draw((0,ymin) -- (0,ymax), arrow=axisarrow);
label("$x$",(xmax,0),E);
label("$y$",(0,ymax),N);
real x = -1.5;
real dx = 1;
real t0 = times(s,x)[0];
real t1 = times(s,x+dx)[0];
path striptop = subpath(s,t0,t1);
filldraw(striptop -- (x+dx,0) -- (x,0) -- cycle, orange, blue);
```



Añadiendo texto

Hay varias maneras de agregar texto a un diagrama de Asymptote. Probablemente la más sencilla es utilizar la función:

```
void label(Label L, pair position);
```

Este comando agrega un texto L en la posición dada por el par de coordenadas position. Por ejemplo, si se quisiera agregar el texto "Punto A" en la posición (1,1), se podría hacer lo siguiente:

Ejemplo 10.44

```
label("Punto $A$", (1,1));
```

Punto A

Argumentos opcionales de la función label se muestra en la tabla 10.23.

Ejemplo 10.45

```
label("Ejemplo", align=W);
```

Ejemplo

Ejemplo 10.46

```
label("Ejemplo", p=red+2bp);
```

Ejemplo

Ejemplo 10.47

```
label("Ejemplo", filltype=FillDraw(yellow, black));
```

Ejemplo

Ejemplo 10.48

```
label("Ejemplo", align=NE, p=blue+1bp, filltype=Fill(yellow));
```

Ejemplo

En Asymptote, también es posible agregar etiquetas directamente a los trazados de las rutas cuando se dibujan, utilizando el parámetro opcional "`Label L`" del comando `draw()`. Cuando se utiliza esta opción, generalmente es necesario construir explícitamente una etiqueta usando la función `Label()`.

```
Label Label(string s);
```

Esta función toma una cadena como argumento y devuelve una etiqueta que puede ser usada como el parámetro L del comando `draw()`. Por ejemplo, si se desea agregar la etiqueta "Punto A" a un punto en las coordenadas (1, 1), se podría hacer lo siguiente:

Ejemplo 10.49

```
unitsize(0,1.5cm);
pair A = (0,0);
draw(A--(3,3), L=Label("Punto A"));
```

Punto A

Parámetros opcionales se puede apreciar en la tabla 10.24.

Ejemplo 10.50

```
size(4cm);
import graph;
```

```
real f(real x) {return sin(x);}
draw(graph(f, -pi, pi), red+1bp, L=Label("$f(x)$", position=MidPoint, align=2*E,
↪  p=fontsize(10pt)));
```


Valores de parámetro `position`:

1. `EndPoint`
2. `MidPoint`
3. `BeginPoint`
4. `Relative(r)`. Valor de $r \in [0,1]$.

Ejemplo 10.51

```
settings.outformat = "pdf";
size(4cm, 0);
real r = 0.2;
Label Lx= Label("$x$", position=EndPoint);
Label Ly = Label("$y$", position=BeginPoint);
Label Lz = Label("$z$", position=MidPoint);
Label Lr = Label("$r$", position=Relative(r), align=SE);
draw((0,0) -- (0,1.5), arrow=Arrow(TeXHead),L=Lx);
draw((1,0) -- (1,1.5), arrow=Arrow(TeXHead),L=Ly);
draw((2,0) -- (2,1.5), arrow=Arrow(TeXHead),L=Lz);
draw((3,0) -- (3,1.5), arrow=Arrow(TeXHead),L=Lr);
```


Ejemplo 10.52

```
settings.outformat="pdf";
unitsize(1cm);
defaultpen(fontsize(10pt));
Label L = Label("$f(x)$", align=(0,0), position=MidPoint, filltype=Fill(white));
draw((0,0) -- (0,1.5), L=L, arrow=Arrows(),bar=Bars);
```

Añadiendo múltiples etiquetas en un path

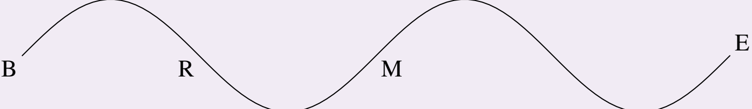
Para añadir múltiples etiquetas a un mismo camino en Asymptote, se deben construir las etiquetas por separado utilizando la función Label. Luego, se puede aplicar el método

```
label(Label L, path g)
```

para añadir cada etiqueta al camino deseado. Este método toma dos argumentos: el objeto Label que se desea añadir y el camino al cual se le quiere aplicar la etiqueta.

Ejemplo 10.53

```
settings.outformat="pdf";
import graph;
size(12cm,1.5cm);
real f(real x){return sin(x);}
path p =graph(f, -2pi, 2pi);
draw(p);
Label L1 = Label("B", position=BeginPoint);
Label L2 = Label("M", position=MidPoint);
Label L3 = Label("E", position=EndPoint);
Label L4 = Label("R", position=Relative(0.25));
label(L1, p);
label(L2, p);
label(L3, p);
label(L4, p);
```



10.3 Gráficas en tres dimensiones

Hello Sphere

Para realizar gráficas en 3d generalmente se utiliza la librería three.

Ejemplo 10.54

```
settings.prc=false;
size(1cm);
import three;
draw(unitsphere);
```

Trazado de líneas en 3 dimensiones

Las líneas en 3 dimensiones se realizan de manera similar que en el paquete tikz definiendo los puntos de la forma (a,b,c).

Ejemplo 10.55

```
settings.prc = false;
import three;
size(4cm,0);
draw((0,0,0)--(2,0,0), blue);//x-axis
draw((0,0,0)--(0,2,0), green); //y-axis
draw((0,0,0)--(0,0,2), red); //z-axis
```

Las definiciones de los vectores $\vec{0}$, $\hat{\imath}$, $\hat{\jmath}$ y $\hat{k}$ se muestran en la tabla 10.25.

Entonces, el código de la tabla es equivalente a:

```
draw(O--2X, blue); //x-axis
draw(O--2Y, green); //y-axis
draw(O--2Z, red); //z-axis
```

Resolución de imágenes

Para mejorar la resolución de las imágenes se puede utilizar el comando

```
settings.outformat="png";
settings.render=16;
```

Paths

Asymptote es un lenguaje de programación para la creación de gráficos vectoriales que cuenta con tipos incorporados para la representación de rutas en tres dimensiones. La sintaxis para la creación de estas rutas utiliza los operadores `--` y `..`, y se puede utilizar la función `dot(triple)` para marcar puntos específicos. Además, el tipo `guide3` es un tipo intermedio que se puede utilizar en lugar de `path3`, y asymptote puede realizar la conversión automáticamente. La palabra clave `cycle` se utiliza para cerrar la curva.

Ejemplo 10.56

```
settings.render=8;
import three;
size(3cm,0);
draw(-X -- X .. Y .. X-Y+Z .. cycle);
dot(-X ^^ X ^^ Y ^^ X-Y+Z);
```

El operador --- es similar al operador -- en que ambos dibujan segmentos de línea entre dos puntos. Sin embargo, el operador --- trata de mantener la suavidad de la ruta más allá de esos puntos, lo que puede ser útil en la creación de curvas suaves. Este operador también funciona en el contexto bidimensional, lo que significa que se puede utilizar tanto en rutas de dos dimensiones como de tres dimensiones.

Ejemplo 10.57

```
settings.render=8;
import three;
size(3cm,0);
draw(-X --- X .. Y .. X-Y+Z .. cycle);
dot(-X ^^ X ^^ Y ^^ X-Y+Z);
```

Plano

La función path3 plane de Asymptote es una función que permite dibujar un plano en un sistema de coordenadas tridimensional (3D), a partir de los vectores que definen los lados del plano y su punto de origen.

La sintaxis de la función es la siguiente:

```
path3 plane(triple u, triple v, triple O=O);
```

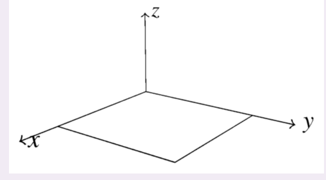
Donde:

1. `triple u` y `triple v` son dos vectores que definen los lados del plano. Estos vectores deben ser perpendiculares entre sí para que el plano sea rectángular.
2. `triple O` es el punto de origen del plano. Este punto es el punto en el espacio tridimensional desde el cual se va a dibujar el plano. El valor predeterminado de O es el origen `(0,0,0)` del sistema de coordenadas 3D.

Ejemplo 10.58

```
size(4cm,0);
import three;
// Definir los vectores que definen los lados del plano
```

```
triple u=(1.5,0,0);
triple v=(0,1.5,0);
// Definir el punto de origen del plano
triple O=(0,0,0);
// Crear el plano
path3 plane1 = plane(u,v,O);
// Dibujar los ejes cartesianos
draw(O--2X, arrow=Arrow3(TeXHead2), L=Label("$x$", position=EndPoint));
draw(O--2Y, arrow=Arrow3(TeXHead2), L=Label("$y$", position=EndPoint));
draw(O--Z, arrow=Arrow3(TeXHead2), L=Label("$z$", position=EndPoint));
// Dibujar el plano
draw(plane1);
```



Caja

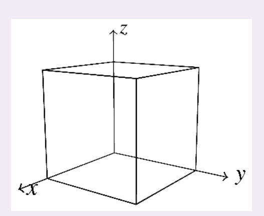
La función "box" en Asymptote se utiliza para dibujar un cubo o caja en el espacio tridimensional. La sintaxis de la función es la siguiente:

```
path3[] box(triple a, triple b);
```

En donde `triple a` y `tripe b` son los vértices opuestos de la caja que se desea dibujar.

Ejemplo 10.59

```
settings.render=16;
import three;
size(3cm,0);
triple b=(1.5,1.5,1.5);
triple a=(0,0,0);
path3[] box1 = box(a,b);
draw(O--2X, arrow=Arrow3(TeXHead2), L=Label("$x$", position=EndPoint));
draw(O--2Y, arrow=Arrow3(TeXHead2), L=Label("$y$", position=EndPoint));
draw(O--2Z, arrow=Arrow3(TeXHead2), L=Label("$z$", position=EndPoint));
draw(box1);
```



Círculo

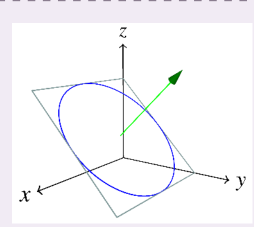
La función "circle" en Asymptote se utiliza para dibujar un círculo en el espacio tridimensional. La sintaxis de la función es la siguiente:

```
path3 circle(triple c, real r, triple normal=Z);
```

1. `triple c` es el centro del círculo en el espacio tridimensional, definido como una terna ordenada de números `(x,y,z)`.
2. `real r` es el radio del círculo.
3. `triple normal` es un vector normal al plano del círculo. Este parámetro es opcional y su valor predeterminado es `(0,0,1)`, que es el vector normal al plano XY.

Ejemplo 10.60

```
settings.render=16;
import three;
size(3cm,0);
draw(O--2X, arrow=Arrow3(TeXHead2), L=Label("$x$", position=EndPoint));
draw(O--2Y, arrow=Arrow3(TeXHead2), L=Label("$y$", position=EndPoint));
draw(O--2Z, arrow=Arrow3(TeXHead2), L=Label("$z$", position=EndPoint));
triple c = (Y+Z)/sqrt(2) + X;
path3 circle1 = circle(c=c, r=1, normal=Y+Z);
draw(plane(O=sqrt(2)*Z, 2X, 2*unit(Y-Z)), gray + 0.1cyan);
draw(circle1, blue);
draw(shift(c) * (O -- Y+Z), green, arrow=Arrow3());
```



Arco

La función "arc" en Asymptote se utiliza para dibujar un arco en el espacio tridimensional. La sintaxis de la función es la siguiente:

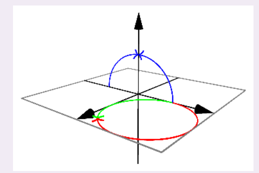
```
path3 arc(triple c, triple v1, triple v2, triple normal=O);
```

Donde:

1. `triple c` es el centro del arco en el espacio tridimensional, definido como una terna ordenada de números `(x,y,z)`.
2. `triple v1` y `triple v2` son dos puntos en el espacio que determinan los extremos del arco. El arco se dibujará en el sentido de las manecillas del reloj desde `v1` hasta `v2`.
3. `triple normal` es un vector normal al plano del arco. Este parámetro es opcional y su valor predeterminado es `(0,0,0)`, lo que significa que la función calculará automáticamente un vector normal al plano del arco.

Ejemplo 10.61

```
settings.render=16;
import three;
size(3cm,0);
draw(-2X--2X,arrow=Arrow3(emissive(black)));
draw(-2Y--2Y, arrow=Arrow3(emissive(black)));
draw(-2Z--2Z, arrow=Arrow3(emissive(black)));
draw(path3(box((-2,-2),(2,2))), gray);
draw(arc(c=O, Y, Z), blue, arrow = Arrow3(TeXHead2, emissive(blue)));
draw(arc(c=O, -Y, Z), blue, arrow = Arrow3(TeXHead2, emissive(blue)));
draw(arc(c=(1,1,0), Y, 2X, normal=Z), green, arrow = Arrow3(TeXHead2(normal=Z),
↪ emissive(green)));
draw(arc(c=(1,1,0), Y, 2X, normal=-Z), red, arrow = Arrow3(TeXHead2(normal=Z),
↪ emissive(red)));
```



Curvas

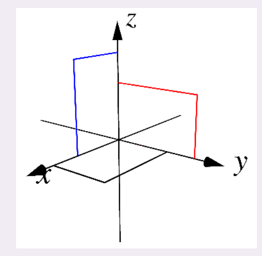
Para convertir una variable de trayectoria bidimensional en una trayectoria tridimensional (path3), se utiliza la función

```
path3 path3(path p, triple plane(pair) = XYplane);
```

Por defecto, esta función mapea la trayectoria en el plano xy aplicando la transformación `(x,y) -> (x,y,0)`. Si se cambia el argumento opcional a ZXplane o YZplane, entonces `(x,y)` se mapea a `(y,0,x)` o `(0,x,y)`, respectivamente.

Ejemplo 10.62

```
settings.render=16;
import three;
size(3cm, 0);
draw(-2X--2X,arrow=Arrow3(emissive(black)), L=Label("$x$", position=EndPoint));
draw(-2Y--2Y, arrow=Arrow3(emissive(black)), L=Label("$y$", position=EndPoint));
draw(-2Z--2Z, arrow=Arrow3(emissive(black)), L=Label("$z$", position=EndPoint));
path p = box((0,0), (3,1));
draw(path3(p), black);
draw(path3(p, plane=ZXplane), blue);
draw(path3(p, plane=YZplane), red);
```



Ejemplo 10.63

```
settings.render = 16;
import graph;
import three;
size(3cm, 0);
real xmin = -0.1;
real xmax = 2;
real ymin = -0.1;
real ymax = 2;
real f(real x) { return sqrt(x); }
path s = graph(f, 0, 2, operator..);
draw(path3(s));
draw(xmin*X--xmax*X,arrow=Arrow3(emissive(black)), L=Label("$x$", position=EndPoint));
draw(ymin*Y--ymax*Y, arrow=Arrow3(emissive(black)), L=Label("$y$", position=EndPoint));
```

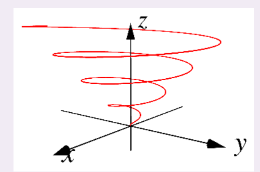


Curvas paramétricas

Para graficar una función paramétrica, primeramente podemos definir la función y luego dibujas las gráficas a través de la función `graph` del módulo `graph3`.

Ejemplo 10.64

```
settings.render=16;
import graph3;
size(3cm, 0);
draw(-8pi*X--8pi*X,arrow=Arrow3(emissive(black)), L=Label("$x$", position=EndPoint));
draw(-8pi*Y--8pi*Y, arrow=Arrow3(emissive(black)), L=Label("$y$", position=EndPoint));
draw(-2pi*Z--8pi*Z, arrow=Arrow3(emissive(black)), L=Label("$z$", position=EndPoint));
triple f(real t) {return (t*cos(t), t*sin(t), t);}
path3 spiral = graph(f, 0, 8pi, operator ..);
draw(spiral, red);
```



Surfaces of revolution

La función "surface" en Asymptote es una función que permite crear superficies tridimensionales en el espacio. La función toma como entrada tres argumentos: un triple "c", una trayectoria tridimensional "g" y otro triple "axis".

```
surface(triple c, path3 g, triple axis)
```

El primer argumento, "c", es un triple que especifica el centro de la superficie. Este triple representa las coordenadas x, y, z del centro de la superficie en el espacio tridimensional.

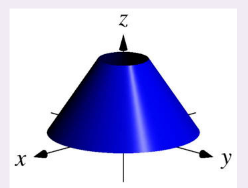
El segundo argumento, "g", es una trayectoria tridimensional que define la forma de la superficie. La trayectoria "g" se puede construir utilizando funciones como "path3", o se puede generar mediante funciones paramétricas. La trayectoria "g" define la forma de la superficie en términos de sus puntos.

El tercer argumento, "axis", es un triple que define la orientación de la superficie. Este triple representa las coordenadas x, y, z de los ejes de la superficie en el espacio tridimensional. Por ejemplo, si se desea que la

superficie esté orientada de manera que su eje x se alinee con el eje z del espacio tridimensional, se puede establecer el valor del tercer argumento como "(0,0,1)".

Ejemplo 10.65

```
import graph3;
import three;
size(3cm, 0);
currentprojection = orthographic(1,1,0.4);
draw(-2*X--2.5*X,arrow=Arrow3(emissive(black)), L=Label("$x$", position=EndPoint));
draw(-2*Y--2.5*Y, arrow=Arrow3(emissive(black)), L=Label("$y$", position=EndPoint));
draw(-1*Z--2*Z, arrow=Arrow3(emissive(black)), L=Label("$z$", position=EndPoint));
path3 segment = (0,1.5,0) -- (0,0.5,1.5);
surface shade = surface(c=0, segment, axis = Z);
draw(shade,blue);
```



Parámetros opcionales se puede observar en la tabla 10.26.

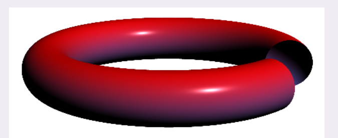
Tabla 10.26

type	name	default value
int	n	nslice (usually 12)
real	angle1	0
real	angle2	360
pen(int,real)	color	null

Ejemplo 10.66

```
settings.render=16;
import graph3;
size(4cm,0);
currentprojection = perspective(30*dir(75,0));
real r1=5, r0=1;
int nu = 36, nv = 36;
path3 crossSection = Circle(r=r0, c=(r1,0,0), normal=Y, n= nu);
pen colorFunction(int u, real theta) {
  real z = sin(u/nu * 2pi);
  real t = (z + 1) / 2;
  return t*red + (1-t)*lightblue;
}
```

```
surface torus = surface(crossSection, c=(0,0,0), axis=Z, n=nv,angle1=90, angle2=410,
↪  color=colorFunction);
draw(torus);
```



Punto de visualización

Para elegir un punto de vista en Asymptote, se puede establecer el valor de la variable “currentprojection” con una de las opciones predefinidas de perspectiva.

1. Perspectiva: La imagen se dibuja con perspectiva, lo que significa que los objetos más cercanos aparecen más grandes que los objetos más lejanos.

2. Ortográfico: hace que la imagen se dibuje sin perspectiva, de manera que los objetos mantienen su tamaño relativo independientemente de su distancia.

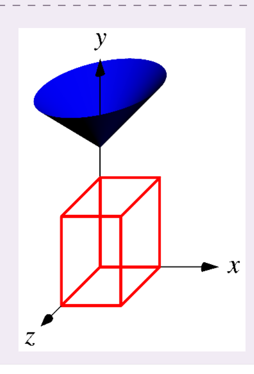
3. Oblicuo: permite especificar un ángulo de inclinación para el plano de proyección, lo que puede crear una vista diagonal de los objetos.

Proyección oblique

Asymptote ofrece proyecciones `obliqueX` y `obliqueY` con los ejes x e y saliendo de la página, y la proyección `obliqueZ` es la misma que la proyección `oblicua` regular.

Ejemplo 10.67

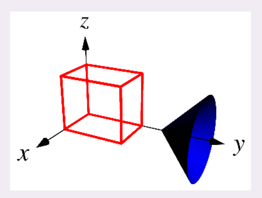
```
settings.render=16;
defaultpen(fontsize(10pt));
import three;
size(3cm,0);
currentprojection = oblique;
draw(O--2*X,arrow=Arrow3(emissive(black)), L=Label("$x$", position=EndPoint));
draw(O--3.5*Y, arrow=Arrow3(emissive(black)), L=Label("$y$", position=EndPoint));
draw(O--2*Z, arrow=Arrow3(emissive(black)), L=Label("$z$", position=EndPoint));
draw(box(O, (1,1.5,1.3)), red+linewidth(1pt));
draw(surface(2Y -- 3Y+X, c=O, axis=Y), blue);
```



Perspectiva

Ejemplo 10.68

```
settings.render=16;
defaultpen(fontsize(10pt));
import three;
size(3cm,0);
currentprojection = perspective(4,3,2);
draw(O--2*X,arrow=Arrow3(emissive(black)), L=Label("$x$", position=EndPoint));
draw(O--3.5*Y, arrow=Arrow3(emissive(black)), L=Label("$y$", position=EndPoint));
draw(O--2*Z, arrow=Arrow3(emissive(black)), L=Label("$z$", position=EndPoint));
draw(box(O, (1,1.5,1.3)), red+linewidth(1pt));
draw(surface(2Y -- 3Y+X, c=O, axis=Y), blue);
```

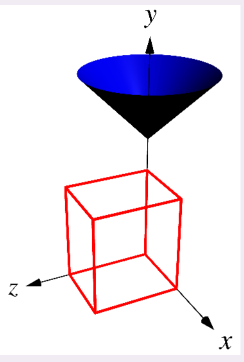


El argumento opcional `triple up` le indica a Asymptote que rote la cámara para que el vector especificado aparezca apuntando hacia arriba en la imagen final, lo que puede ser útil para asegurar una orientación coherente de la escena.

Ejemplo 10.69

```
settings.render=16;
defaultpen(fontsize(10pt));
import three;
size(3cm,0);
currentprojection = perspective(4,3,2);
draw(O--2*X,arrow=Arrow3(emissive(black)), L=Label("$x$", position=EndPoint));
```

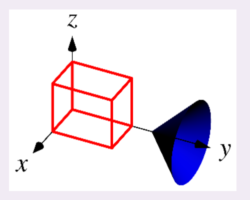
```
draw(O--3.5*Y, arrow=Arrow3(emissive(black)), L=Label("$y$", position=EndPoint));
draw(O--2*Z, arrow=Arrow3(emissive(black)), L=Label("$z$", position=EndPoint));
draw(box(O, (1,1.5,1.3)), red+linewidth(1pt));
draw(surface(2Y -- 3Y+X, c=O, axis=Y), blue);
```



Proyección orthographic

Ejemplo 10.70

```
settings.render=16;
defaultpen(fontsize(10pt));
import three;
size(3cm,0);
currentprojection = orthographic;
draw(O--2*X,arrow=Arrow3(emissive(black)), L=Label("$x$", position=EndPoint));
draw(O--3.5*Y, arrow=Arrow3(emissive(black)), L=Label("$y$", position=EndPoint));
draw(O--2*Z, arrow=Arrow3(emissive(black)), L=Label("$z$", position=EndPoint));
draw(box(O, (1,1.5,1.3)), red+linewidth(1pt));
draw(surface(2Y -- 3Y+X, c=O, axis=Y), blue);
```



También se puede especificar la orientación de la cámara con el argumento opcional `triple up`.

Sólidos predefinidos

Los sólidos predefinidos son:

1. unitsphere: esfera unitaria.
2. unitdisk: disco unitario.
3. unitplane: plano unitario.
4. unitcube: cubo unitario.
5. unitcylinder: cilindro unitario.
6. unitcone: cono unitario.
7. unitsolidcone: cono sólido unitario.
8. unithemisphere: hemisferio unitario.

Ejemplo 10.71

```
settings.render=16;
import three;
size(1cm,0);
draw(unitsphere, surfacepen=material(white,
↪ emissivepen=gray(0.3)));
```

Transformaciones en tres dimensiones

las superficies predefinidas son limitadas, pero se pueden hacer mucho más flexibles usando transformaciones tridimensionales. Las transformaciones tridimensionales tienen el tipo transform3 y pueden aplicarse y componerse mediante el operador ∗. Algunas transformaciones útiles incluyen shift, scale3, rotate, reflect y shear, y pueden combinarse para producir formas más complejas y personalizadas.

Ejemplo 10.72

```
settings.render=16;
import three;
size(3cm,0);
draw( scale(1,1,2) * shift(3X) * rotate(180, Y) *
↪ unitcone, surfacepen=white);
draw(shift(3,0,0) * unitcylinder, surfacepen = white);
draw(shift(3,0,1) * unitdisk, surfacepen =
↪ emissive(blue));
```

Otros funciones se muestra en la tabla 10.27.

Tabla 10.27

function	resultado de la transformación
`scale3(real r)`	factor de escala r
`scale(real a, real b, real c)`	escala por a el x, b por y y c por z
`scale(triple t)`	equivalente a `scale(t.x, t.y, t.z)`
`xscale3(real r)`	equivalente a `scale(r, 1, 1)`
`yscale3(real r)`	equivalente a `scale(1, r, 1)`
`zscale3(real r)`	equivalente a `scale(1, 1, r)`

Las reflexiones en 3 dimensiones se pueden realizar con la función `reflect` que toma un argumento opcional `triple axis` que especifica el eje de reflexión (ver tabla 10.28).

función	efecto
`reflect(O,Z,X+Y)`	intercambia x y y
`reflect(O,Y,X+Z)`	intercambia x y z
`reflect(O,X,Y+Z)`	intercambia y y z

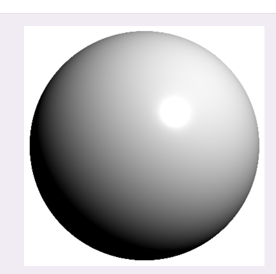
Tabla 10.28

Iluminación

Un objeto se puede ver oscura cuando la luz está golpeando en un ángulo desfavorable por lo que es necesario adoptar un ángulo mejor para su iluminación. La opción predeterminada de light es nolight y además existen otros valores como viewport, white y headlamp.

Ejemplo 10.73

```
settings.outformat = "pdf";
settings.render = 16;
import graph;
import three;
size(3cm);
draw(unitsphere,white,light=White);
```



10.4 Programación asymptote

Comandos más comunes de programación

```
void draw(picture pic=currentpicture, Label L="", path g,
        align align=NoAlign, pen p=currentpen,
        arrowbar arrow=None, arrowbar bar=None, margin margin=NoMargin,
        Label legend="", marker marker=nomarker);

frame legend(picture pic=currentpicture, int perline=1,
           real xmargin=legendmargin, real ymargin=xmargin,
           real linelength=legendlinelength,
           real hskip=legendhskip, real vskip=legendvskip,
           real maxwidth=0, real maxheight=0,
           bool hstretch=false, bool vstretch=false, pen p=currentpen);

void dot(frame f, pair z, pen p=currentpen, filltype filltype=dotfilltype);
void dot(picture pic=currentpicture, pair z, pen p=currentpen,
         filltype filltype=dotfilltype);
void dot(picture pic=currentpicture, Label L, pair z, align align=NoAlign,
         string format=defaultformat, pen p=currentpen, filltype
↪ filltype=dotfilltype);
void dot(picture pic=currentpicture, Label[] L=new Label[], pair[] z,
```

```
          align align=NoAlign, string format=defaultformat, pen p=currentpen,
          filltype filltype=dotfilltype);
void dot(picture pic=currentpicture, path[] g, pen p=currentpen,
          filltype filltype=dotfilltype);
void dot(picture pic=currentpicture, Label L, pen p=currentpen,
          filltype filltype=dotfilltype);

void dot(picture pic=currentpicture, Label[] L=new Label[],
        explicit path g, align align=RightSide, string format=defaultformat,
        pen p=currentpen, filltype filltype=dotfilltype);

void draw(pair origin, picture pic=currentpicture, Label L="", path g,
           align align=NoAlign, pen p=currentpen, arrowbar arrow=None,
           arrowbar bar=None, margin margin=NoMargin, Label legend="",
           marker marker=nomarker);

void fill(picture pic=currentpicture, path g, pen p=currentpen);

void filldraw(picture pic=currentpicture, path g, pen fillpen=currentpen,
            pen drawpen=currentpen);

void fill(pair origin, picture pic=currentpicture, path g, pen p=currentpen);

void filloutside(picture pic=currentpicture, path g, pen p=currentpen);

void latticeshade(picture pic=currentpicture, path g, bool stroke=false,
                pen fillrule=currentpen, pen[][] p)

void axialshade(picture pic=currentpicture, path g, bool stroke=false,
              pen pena, pair a, bool extenda=true,
              pen penb, pair b, bool extendb=true);

void radialshade(picture pic=currentpicture, path g, bool stroke=false,
               pen pena, pair a, real ra, bool extenda=true,
               pen penb, pair b, real rb, bool extendb=true);

void gouraudshade(picture pic=currentpicture, path g, bool stroke=false,
                pen fillrule=currentpen, pen[] p, pair[] z,
                int[] edges);
void gouraudshade(picture pic=currentpicture, path g, bool stroke=false,
                  pen fillrule=currentpen, pen[] p, int[] edges);
```

```
void tensorshade(picture pic=currentpicture, path[] g, bool stroke=false,
               pen fillrule=currentpen, pen[][] p, path[] b=g,
               pair[][] z=new pair[][]);

void tensorshade(picture pic=currentpicture, path g, bool stroke=false,
               pen fillrule=currentpen, pen[] p, path b=g,
               pair[] z=new pair[]);

void draw(picture pic=currentpicture, pen fillrule=currentpen, path[] g,
        pen[] p);

void functionshade(picture pic=currentpicture, path[] g, bool stroke=false,
                 pen fillrule=currentpen, string shader);

void unfill(picture pic=currentpicture, path g);

void clip(picture pic=currentpicture, path g, stroke=false,
        pen fillrule=currentpen);

void label(picture pic=currentpicture, Label L, pair position,
         align align=NoAlign, pen p=currentpen, filltype filltype=NoFill)

Label Label(string s="", pair position, align align=NoAlign,
          pen p=nullpen, embed embed=Rotate, filltype filltype=NoFill);
Label Label(string s="", align align=NoAlign,
            pen p=nullpen, embed embed=Rotate, filltype filltype=NoFill);
Label Label(Label L, pair position, align align=NoAlign,
            pen p=nullpen, embed embed=L.embed, filltype filltype=NoFill);
Label Label(Label L, align align=NoAlign,
            pen p=nullpen, embed embed=L.embed, filltype filltype=NoFill);

void label(picture pic=currentpicture, Label L, path g, align align=NoAlign,
         pen p=currentpen, filltype filltype=NoFill);

void arrow(picture pic=currentpicture, Label L="", pair b, pair dir,
         real length=arrowlength, align align=NoAlign,
         pen p=currentpen, arrowbar arrow=Arrow, margin margin=EndMargin);

object draw(picture pic=currentpicture, Label L, envelope e,
          real xmargin=0, real ymargin=xmargin, pen p=currentpen,
          filltype filltype=NoFill, bool above=true);
object draw(picture pic=currentpicture, Label L, envelope e, pair position,
            real xmargin=0, real ymargin=xmargin, pen p=currentpen,
            filltype filltype=NoFill, bool above=true);
```

Secuencias de control

Comentario:

```
// This is a comment.
```

Declaración de variable:

```
real x;
```

Asignación de variable:

```
x=1.0;
```

Condicional:

```
if(x == 1.0) {
  write("x equals 1.0");
} else {
  write("x is not equal to 1.0");
}
```

Loop:

```
for(int i=0; i < 10; ++i) {
  write(i);
}
```

Soporta `while`, `do`, `break` y `continue` como en C/C++.

Iteración de array:

```
int[] array={1,1,2,3,5};
for(int k : array) {
  write(k);
}
```

Tipos de datos

Los tipos de datos son:

1. `void`. El tipo void se utiliza solo por las funciones que no reciben argumentos o que no devuelven ningún valor.

2 `bool`. El tipo bool se utiliza para representar valores booleanos. Los valores posibles son `true` y `false`.

3 `bool3`. Un tipo booleano extendido que puede tomar los valores true, default o falso. Un tipo bool3 puede convertirse desde o hacia un bool. El inicializador predeterminado para bool3 es "default".

4 `int`. Este es un tipo de dato entero. Si no se le da un valor inicial, se asume que su valor es 0. El valor mínimo permitido para este tipo de dato es intMin y el valor máximo es intMax.

5 `real`. Este es un tipo de dato que representa un número real. Debe ser configurado para el tipo de punto flotante nativo de la arquitectura con la mayor precisión posible. El valor inicial implícito para los números reales es 0.0. Estos números tienen una precisión de realEpsilon y realDigits dígitos significativos. El número real positivo más pequeño que se puede representar es realMin y el más grande es realMax. Las variables inf y nan, junto con la función bool isnan(real x), son útiles cuando se enmascaran las excepciones de punto flotante con la opción de línea de comando -mask (que es la opción predeterminada en modo interactivo).

6 `pair`. Este es un número complejo, es decir, un par ordenado de componentes reales `(x,y)`. Las partes real e imaginaria de un par `z` se pueden leer como `z.x` y `z.y`. Decimos que `x` y `y` son miembros virtuales del par de elementos de datos; sin embargo, no se pueden modificar directamente. El inicializador implícito para los pares es `(0.0,0.0)`. Determinando el conjugado del número complejo:

```
pair z = (1,2);
z = (z.x, -z.y);
z = z.x-I*z.y;
z = conj(z);
```

- `pair conj(pair z)`. Devuelve el conjugado del número complejo `z`.
- `real length(pair z)`. Devuelve la longitud del número complejo `z`.
- `real angle(pair z, bool warn=true)`. Devuelve el ángulo del número complejo `z` en radianes.
- `real degrees(pair z, bool warn=true)`. Devuelve el ángulo del número complejo `z` en grados en el intervalo `[0,360)`. Un par `z=(0,0)` produce error.
- `pair unit(pair z)`. Devuelve un vector unitario en la dirección del número complejo `z`.
- `pair expi(real angle)`. Devuelve un vector unitario en la dirección del ángulo `angle` medido en radianes.
- `pair dir(real degrees)`. Devuelve un vector unitario en la dirección del ángulo `degrees` medido en grados.
- `real xpart(pair z)`. Devuelve `z.x`.
- `real ypart(pair z)`. Devuelve `z.y`.
- `pair realmult(pair z, pair w)`. Devuelve el producto elemento a elemento `(z.x*w.x,z.y*w.y)`.

- `real dot(explicit pair z, explicit pair w)`. Devuelve el producto punto `z.x*w.x+z.y*w.y`.
- `real cross(explicit pair z, explicit pair w)`. Devuelve el producto cruz `z.x*w.y-z.y*w.x`.
- `real orient(pair a, pair b, pair c)`. Devuelve un valor positivo (negativo) si `a--b--c--cycle` está orientado en sentido antihorario (horario) o cero si todos los tres puntos son colineales. Equivalente, un valor positivo (negativo) es devuelto si `c` está a la izquierda (derecha) de la línea que pasa por `a` y `b` o cero si `c` está en esta línea. El valor devuelto puede ser expresado en términos del producto cruz 2D como `cross(a-c,b-c)`, que es el determinante

$$\begin{vmatrix} a.x & a.y & 1 \\ b.x & b.y & 1 \\ c.x & c.y & 1 \end{vmatrix}$$

- `real incircle(pair a, pair b, pair c, pair d)`. Devuelve un valor positivo (negativo) si `d` está dentro (fuera) del círculo que pasa por los puntos orientados en sentido antihorario `a,b,c` o cero si `d` está en este círculo. El valor devuelto es el determinante

$$\begin{vmatrix} a.x & a.y & a.x^2+a.y^2 & 1 \\ b.x & b.y & b.x^2+b.y^2 & 1 \\ c.x & c.y & c.x^2+c.y^2 & 1 \\ d.x & d.y & d.x^2+d.y^2 & 1 \end{vmatrix}$$

- `pair minbound(pair z, pair w)`. Devuelve `(min(z.x,w.x),min(z.y,w.y))`.
- `pair maxbound(pair z, pair w)`. Devuelve `(max(z.x,w.x),max(z.y,w.y))`.

7 `trple` es un triple ordenado `(x,y,z)` de componentes reales usados para dibujos tridimensionales. Los componentes de un triple `v` se pueden leer como `v.x`, `v.y` y `v.z`. El inicializador implícito para triples es `(0.0,0.0,0.0)`.

- `real length(triple v)` regresa la longitud `|v|` de su argumento `v`. Un sinónimo para `length(triple)` es `abs(triple)`. La función `abs2(triple v)` regresa `|v|^2`.
- `real polar(triple v, bool warn=true)` regresa la colatitud de `v` medida desde el eje `z` en radianes o `0` si `warn` es `false` y `v=O` (en lugar de producir un error).
- `real azimuth(triple v, bool warn=true)` regresa la longitud de `v` medida desde el eje `x` en radianes o `0` si `warn` es `false` y `v.x=v.y=0` (en lugar de producir un error).
- `real colatitude(triple v, bool warn=true)` regresa la colatitud de `v` medida desde el eje `z` en grados o `0` si `warn` es `false` y `v=O` (en lugar de producir un error).
- `real latitude(triple v, bool warn=true)` regresa la latitud de `v` medida desde el plano `xy` en grados o `0` si `warn` es `false` y `v=O` (en lugar de producir un error).
- `real longitude(triple v, bool warn=true)` regresa la longitud de `v` medida desde el eje `x` en grados o `0` si `warn` es `false` y `v.x=v.y=0` (en lugar de producir un error).

- triple unit(triple v) regresa el vector unitario en la dirección de v.
- triple expi(real polar, real azimuth) regresa un vector unitario en la dirección (polar,azimuth) medida en radianes.
- triple dir(real colatitude, real longitude) regresa un vector unitario en la dirección (colatitude,longitude) medida en grados.
- real xpart(triple v) regresa v.x.
- real ypart(triple v) regresa v.y.
- real zpart(triple v) regresa v.z.
- real dot(triple u, triple v) regresa el producto punto u.x*v.x+u.y*v.y+u.z*v.z.
- triple cross(triple u, triple v) regresa el producto cruz (u.y*v.z-u.z*v.y,u.z*v.x-u.x*v.z,u.x*v.y-v.x*u.y).
- triple minbound(triple u, triple v) regresa (min(u.x,v.x),min(u.y,v.y),min(u.z,v.z)).
- triple maxbound(triple u, triple v) regresa (max(u.x,v.x),max(u.y,v.y),max(u.z,v.z)).

8 string Este es un tipo de dato que representa una cadena de caracteres, implementado utilizando la clase string de STL. Las cadenas de caracteres delimitadas por comillas dobles (") están sujetas a las siguientes transformaciones para permitir el uso de comillas dobles en TEX (por ejemplo, para usar el paquete babel):

- \" se transforma a \"
- \\ se transforma a \\
- \` se transforma a `
- \" se transforma a "
- \? se transforma a ?
- \\ se transforma a backslash
- \a se transforma a alert
- \b se transforma a backspace
- \f se transforma a form feed
- \n se transforma a newline
- \r se transforma a carriage return
- \t se transforma a tab
- \v se transforma a vertical tab
- \0-\377 se transforma a corresponding octal byte
- \x0-\xFF se transforma a corresponding hexadecimal byte

La inicialización implícita para cadenas de caracteres es la cadena de caracteres vacía "". Las cadenas de caracteres pueden concatenarse con el operador +. En las siguientes funciones de cadenas de caracteres, la posición 0 denota el inicio de la cadena de caracteres:

- `int length(string s)` devuelve la longitud de la cadena de caracteres `s`.
- `int find(string s, string t, int pos=0)` devuelve la posición de la primera ocurrencia de la cadena de caracteres `t` en la cadena de caracteres `s` en o después de la posición `pos`, o -1 si `t` no es una subcadena de `s`.
- `int rfind(string s, string t, int pos=-1)` devuelve la posición de la última ocurrencia de la cadena de caracteres `t` en la cadena de caracteres `s` en o antes de la posición `pos` (si `pos`=-1, al final de la cadena de caracteres `s`), o -1 si `t` no es una subcadena de `s`.
- `string insert(string s, int pos, string t)` devuelve la cadena de caracteres formada por insertar la cadena de caracteres `t` en la posición `pos` en `s`.
- `string erase(string s, int pos, int n)` devuelve la cadena de caracteres formada por borrar la cadena de caracteres de longitud `n` (si `n`=-1, hasta el final de la cadena de caracteres `s`) en la posición `pos` en `s`.
- `string substr(string s, int pos, int n)` devuelve la subcadena de longitud `n` (si `n`=-1, hasta el final de la cadena de caracteres `s`) en la posición `pos` en `s`.
- `string reverse(string s)` devuelve la cadena de caracteres `s` invertida.
- `string replace(string s, string t, string u)` devuelve la cadena de caracteres `s` con todas las ocurrencias de la cadena de caracteres `t` reemplazadas por la cadena de caracteres `u`.
- `string[] split(string s, string delimiter="")` devuelve un arreglo de cadenas de caracteres obtenido al dividir `s` en subcadenas delimitadas por `delimiter` (un delimitador vacío significa un espacio, pero con delimitadores duplicados descartados).
- `string[] array(string s)` devuelve un arreglo de cadenas de caracteres obtenido al dividir `s` en caracteres individuales. La operación inversa se proporciona por `operator +(...string[] a)`.
- `string format(string s, int n, string locale="")` devuelve una cadena de caracteres que contiene `n` formateado de acuerdo con la cadena de formato de estilo C `s` usando la localidad `locale` (o la localidad actual si se especifica una cadena vacía), siguiendo el comportamiento de la función C `fprintf`), excepto que solo se permite un campo de datos.
- `string format(string s=defaultformat, bool forcemath=false, string s=defaultseparator, real x, string locale="")` devuelve una cadena de caracteres que contiene `x` formateado de acuerdo con la cadena de formato de estilo C `s` usando la localidad `locale` (o la localidad actual si se especifica una cadena vacía), siguiendo el comportamiento de la función C `fprintf`), excepto que solo se permite un campo de datos, los ceros finales se eliminan por defecto (a menos que se especifique #), y si `s` especifica el modo matemático o `forcemath=true`, se usa \TeX para mostrar la notación científica usando el `defaultseparator="\!\times\!";`;

- `int hex(string s)` convierte una cadena de caracteres hexadecimal `s` a un entero;
- `int ascii(string s)` devuelve el código ASCII de la primera letra de la cadena de caracteres `s`;
- `string string(real x, int digits=realDigits)` convierte `x` a una cadena de caracteres usando la precisión `digits` y la localidad C;
- `string locale(string s="")` establece la localidad a la cadena de caracteres `s`, si no está vacía, y devuelve la localidad actual;

Functions

Las funciones en Asymptote se tratan como variables con una firma (las variables no funcionales tienen una firma nula). Se permiten variables con el mismo nombre, siempre y cuando tengan firmas distintas.

Los argumentos de las funciones se pasan por valor. Para pasar un argumento por referencia se debe encerrar en una estructura.

Aquí hay algunas características significativas de las funciones en Asymptote:

1. Las funciones pueden ser recursivas, es decir, una función puede llamarse a sí misma.
2. Las funciones pueden ser anidadas, lo que significa que una función puede estar definida dentro de otra función.
3. Las funciones pueden tener valores de retorno. Si no se especifica un valor de retorno, la función devuelve el valor nulo.
4. Las funciones pueden tener argumentos predeterminados, lo que significa que si no se proporciona un valor para un argumento, se utilizará el valor predeterminado especificado.
5. Las funciones pueden tener argumentos de longitud variable, lo que significa que se pueden pasar diferentes números de argumentos a la función en diferentes llamadas.

Variables y funciones

En Asymptote, las variables con firmas (funciones) y sin firmas (variables no funcionales) son distintas.

```
int x, x();
x=5;
x=new int() {return 17;};
x=x();
```

En el ejemplo proporcionado, se declaran dos variables diferentes llamadas “x”: una es una variable entera y la otra es una función sin argumentos con valor de retorno a un entero.

La línea “`x = 5`” asigna el valor entero 5 a la variable “x” que es una variable entera, no una función.

La línea “`x = new int() {return 17;};`” crea un nuevo objeto de tipo entero usando una asignación dinámica de memoria. El objeto recién creado es una función sin argumentos y con valor de retorno entero. La función devuelve el valor 17 cuando se llama. La variable “x” ahora apunta a esta función.

Finalmente, la línea “`x = x();`” llama a la función “x” y asigna el resultado, que es 17, a la variable entera “x”.

Definición de función de forma tradicional

Definición de funciones de forma tradicional:

```
int sqr(int x)
{
  return x*x;
}
sqr=null;
```

Este código en Asymptote define una función llamada “`sqr`” que toma un argumento “x” de tipo entero y devuelve el resultado de multiplicar “x” por sí mismo. La segunda línea de código, “`sqr=null;`” asigna el valor “`null`” a la variable “`sqr`”, lo que significa que ya no se refiere a la función previamente definida. Esto puede generar un error si se intenta llamar a la función “`sqr`” después de la asignación “`sqr=null;`” ya que “`sqr`” ya no se refiere a la función previamente definida y se considera una variable “`null`”. Otro ejemplo de definición de una función de forma tradicional:

```
pair newfunction(pair z, real r=0)
{
  real a,b;
  a=z.x;
  b=z.y;
  return (a+r,b+r);
}
```

Esta función tiene el nombre newFunction y es de tipo pair. Para llamar a la función debemos de pasar los parámetros (a,b); por ejemplo,

```
draw(newFunction((1,2),2) -- newFunction((6,7),0));
```

Es importante tener en cuenta que la conversión de tipos puede generar errores si se realiza incorrectamente. Por lo tanto, es necesario tener un conocimiento sólido de los diferentes tipos de datos en Asymptote y de cómo funcionan las conversiones de tipos antes de utilizarlas en el código.

```
int a, a(), b, b(); // Valid: creates four variables.
a=b;                // Invalid: assignment is ambiguous.
```

```
a=(int) b;          // Valid: resolves ambiguity.
(int) (a=b);        // Valid: resolves ambiguity.
(int) a=b;          // Invalid: cast expressions cannot be L-values.
int c();
c=a;                // Valid: only one possible assignment.
```

1. `int a, a(), b, b();`: Se están declarando cuatro variables, dos de tipo entero y dos funciones sin argumentos que devuelven un entero.

2. `a=b;`: Esta línea intenta asignar la variable `b` a la variable `a`, pero como hay una función con el mismo nombre que la variable, esto genera una ambigüedad y se produce un error.

3. `a=(int) b;`: En esta línea se está utilizando un casting para convertir la variable `b` a un entero y luego asignar ese valor a la variable `a`. Esto resuelve la ambigüedad y se realiza la asignación correctamente.

4. `(int) (a=b);`: Esta línea utiliza dos expresiones de casting para asignar el valor de la variable `b` a la variable `a`. Primero se realiza la asignación con el operador =, y luego se convierte el resultado a un entero. Esto resuelve la ambigüedad y se realiza la asignación correctamente.

5. `(int) a=b;`: En esta línea se intenta asignar el valor de la variable `b` a la variable `a` y luego convertir ese valor a un entero. Sin embargo, los castings solo pueden ser aplicados a expresiones, no a variables, y por lo tanto se produce un error.

6. `int c(); c=a;`: Se declara una función llamada `c` que devuelve un entero y luego se asigna el valor de la variable `a` a la función `c`. Como no hay ambigüedad en esta línea, la asignación se realiza correctamente.

Funciones anónimas

Se permiten las funciones anónimas (también llamadas “de orden superior”) en el código.

```
typedef int intop(int);
intop adder(int m)
{
  return new int(int n) {return m+n;};
}
intop addby7=adder(7);
write(addby7(1));   // Writes 8.
```

En este ejemplo, se define un tipo de función llamado intop que recibe un entero y devuelve un entero. La función adder recibe un entero m y devuelve una función anónima que recibe otro entero n y devuelve la suma de m y n.

La definición de la función anónima se realiza mediante la palabra clave new seguida del tipo de valor que se devuelve (int) y los parámetros de la función (en este caso, solo n). Dentro de las llaves se escribe el cuerpo de la función, que en este caso es simplemente sumar m y n.

Después de definir la función adder, se crea una nueva variable llamada addby7 que se inicializa con la función adder y se le pasa como argumento el valor 7. Esto significa que addby7 es una función anónima que suma 7 al valor que se le pase como argumento.

Finalmente, se llama a la función write y se le pasa como argumento la expresión addby7(1), que llama a la función anónima almacenada en la variable addby7 con el valor 1. La función anónima suma 7 (el valor inicial de addby7) y 1, lo que devuelve 8. Por lo tanto, write(addby7(1)) escribe 8 en la salida de la consola.

Redefinición de funciones

Es posible redefinir una función f, incluso para llamadas a f en funciones declaradas previamente, asignándole otra función (anónima o con nombre). Sin embargo, si f ha sido sobrecargada con una nueva definición de función, las llamadas previas todavía accederán a la versión original de f, como se ilustra en este ejemplo:

```
void f() {write("hi");}
void g() {f();}
g(); // writes "hi"
f=new void() {write("bye");};
g(); // writes "bye"
void f() {write("overloaded");};
f(); // writes "overloaded"
g(); // writes "bye"
```

Primero, se define una función llamada f que imprime "hi" en la salida de la consola. A continuación, se define otra función llamada g que llama a la función f. Se llama a la función g, lo que escribe "hi" en la salida de la consola.

Luego, se reasigna la función f con una nueva función anónima que imprime "bye" en la salida de la consola. Se llama a la función g nuevamente, lo que escribe "bye" en la salida de la consola.

Después, se define una nueva versión sobrecargada de la función f que imprime "overloaded" en la salida de la consola. Se llama a la función f, lo que escribe "overloaded" en la salida de la consola.

Finalmente, se llama a la función g una vez más, lo que imprime "bye" en la salida de la consola. Esto se debe a que la llamada a f dentro de la función g sigue haciendo referencia a la versión original de f, ya que fue definida antes de la redefinición de f con la nueva función anónima.

Las funciones anónimas se pueden utilizar para redefinir una variable de función que ha sido declarada (y se inicializa implícitamente a la función nula) pero aún no se ha definido explícitamente:

```
void f(bool b);
void g(bool b) {
  if(b) f(b);
  else write(b);
}
f=new void(bool b) {
  write(b);
  g(false);
};
g(true); // Writes true, then writes false.
```

Primero, se declara una función llamada f que recibe un parámetro booleano. Luego, se define otra función llamada g que también recibe un parámetro booleano. Dentro de la función g, se realiza una comprobación condicional y si el valor de b es verdadero, se llama a la función f con el valor de b. De lo contrario (else), se escribe el valor de b en la salida de la consola.

A continuación, se redefine la variable de función f con una nueva función anónima que recibe un parámetro booleano y escribe el valor de b en la salida de la consola.

Luego, se llama a la función g con el valor de true, lo que escribe true en la salida de la consola y llama a la función anónima almacenada en la variable f con el valor de false. La función anónima escribe false en la salida de la consola y llama a la función g con el valor de false. Esto escribe false en la salida de la consola.

Argumento por defecto

El valor del argumento predeterminado se determina evaluando la expresión Asymptote dada en el ámbito donde se define la función llamada.

```
real f(int a=1, real b=0) {return a+b;}
```

Nombre de argumentos

A veces es difícil recordar el orden en que aparecen los argumentos en la declaración de una función. Los argumentos con nombre (también conocidos como argumentos de palabras clave) hacen que llamar a funciones con múltiples argumentos sea más fácil.

A diferencia de los lenguajes C y C++, una asignación en un argumento de función se interpreta como una asignación a un parámetro del mismo nombre en la firma de la función, y no dentro del ámbito local.

La opción de línea de comandos -d se puede usar para verificar el código Asymptote en casos donde un argumento con nombre se puede confundir con una asignación local.

Cuando se trata de emparejar argumentos con la firma de una función, primero se emparejan todas las palabras clave, luego los argumentos sin nombres se emparejan con los formales no emparejados de manera habitual.

```
int f(int x, int y) {
  return 10x+y;
}
write(f(4,x=3));
```

La función `f` devuelve `10x+y`. En este caso, `x` tiene un valor de 3 (ya que se emparejó con el valor 3 en la llamada a la función), mientras que `y` tiene un valor de 4 (ya que se emparejó con el segundo argumento en la llamada a la función). Por lo tanto, el resultado de la función es 34.

En ocasiones, puede ser deseable asignar un valor a una variable local dentro de un argumento de función (generalmente no es una buena práctica de programación). En Asymptote, se puede hacer esto simplemente encerrando la asignación entre paréntesis.

Por ejemplo, dada la definición de la función `f` en el ejemplo anterior, la siguiente llamada a la función:

```
int x;
write(f(4,(x=3)));
```

es equivalente a los siguientes dos statements:

```
int x;
x=3;
write(f(4,x));
```

y produce la salida 43.

Esto ocurre porque la asignación en `(x=3)` se interpreta como una asignación al parámetro `x` de la función `f`, en lugar de como una declaración de una variable local llamada `x`.

10.5 Módulo

Nuestros módulo creados por nosotros debemos guardar en directorio `HOME/.asy/nombreModulo`

Para importar un módulo debemos escribir:

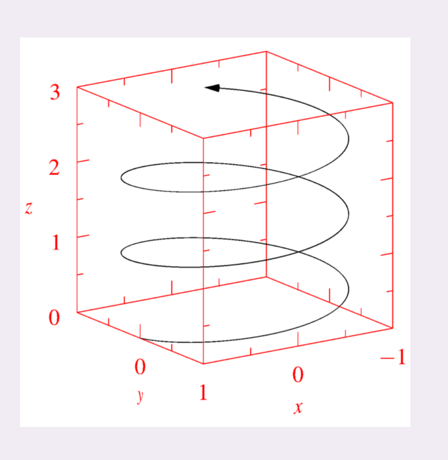
```
import nombreModulo;
```

10.6 Funciones matemáticas

Funciones matemáticas en tres dimensiones

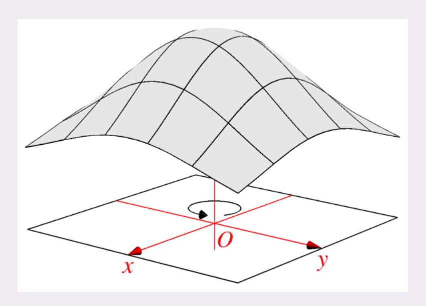
Ejemplo 10.74

```
import graph3;
size(5cm);
size3(200,IgnoreAspect);
currentprojection=orthographic(4,6,3);
real x(real t) {return cos(2pi*t);}
real y(real t) {return sin(2pi*t);}
real z(real t) {return t;}
path3 p=graph(x,y,z,0,2.7,operator ..);
draw(p,Arrow3);
scale(true);
xaxis3(XZ()*"$x$",Bounds,red,InTicks(Label,2,
↪ 2));
yaxis3(YZ()*"$y$",Bounds,red,InTicks(beginlabel
↪ =false,Label,2,2));
zaxis3(XZ()*"$z$",Bounds,red,InTicks);
```



Ejemplo 10.75

```
import graph3;
size(5cm,0);
currentprojection=perspective(10,8,4);
real f(pair z) {return 0.5+exp(-abs(z)^2);}
draw((-1,-1,0)--(1,-1,0)--(1,1,0)--(-1,1,0)--cy
↪ cle);
draw(arc(0.12Z,0.2,90,60,90,25),ArcArrow3);
surface s=surface(f,(-1,-1),(1,1),nx=5,Spline);
xaxis3(Label("$x$"),red,Arrow3);
yaxis3(Label("$y$"),red,Arrow3);
zaxis3(XYZero(extend=true),red,Arrow3);
draw(s,lightgray,meshpen=black+thick(),nolight,
↪ render(merge=true));
label("$O$",O,-Z+Y,red);
```



ÍNDICE Y BIBLIOGRAFÍA

11.1 Generación de índice alfabético

Cuando se escribe un libro extenso, es necesario crear un índice alfabético para que el lector pueda encontrar fácilmente la información que busca. El índice alfabético es una lista de palabras o frases que aparecen en el texto, ordenadas alfabéticamente, con las páginas en las que aparecen. Es una herramienta muy útil tanto para el lector como para el autor, ya que permite comprobar que se han utilizado correctamente las palabras clave en el texto. El índice alfabético se genera automáticamente a partir de las palabras clave utilizadas en el texto. Para que el índice alfabético se genere correctamente, es necesario que las palabras clave estén escritas correctamente en el texto. En este capítulo se explica cómo se genera el índice alfabético y cómo modificar su aspecto.

Diagrama de generación del índice alfabético

Para crear un índice alfabético debemos de recurrir al paquete `makeidx`.

Ejemplo 11.1

```
\documentclass{book}
\usepackage{makeidx}
\usepackage[paperwidth=10cm,paperheight=10cm, margin=1cm]{geometry}
\makeindex
\begin{document}
El comando index crea una entrada para el índice \index{Palabra clave}

El comando printindex imprime el índice alfabético.
\printindex
\end{document}
```

1

El comando index crea una entrada para el índice

El comando printindex imprime el índice alfabético.

Index

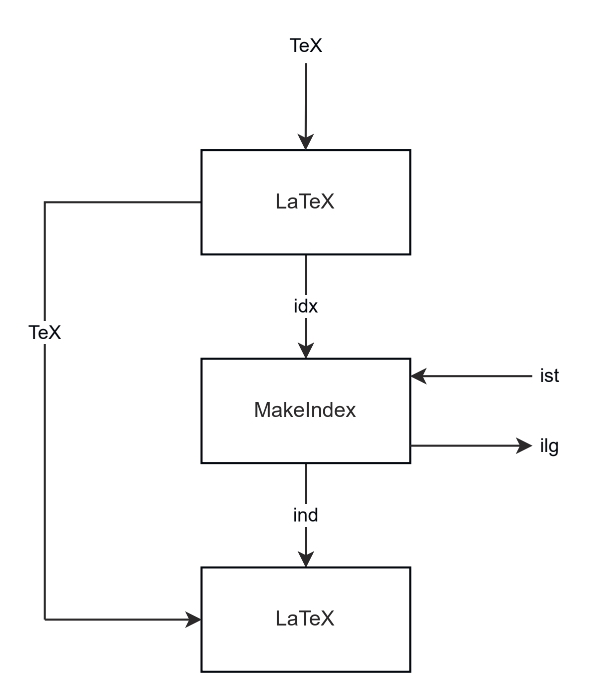
Palabra clave, 1

2

Al momento de compilar un documento LaTeX que tiene índice alfabético debemos compilar con PDFLaTeX - makeindex - PDFLaTeX. Cuando se compile primeramente en PDFLaTeX se generará todas las entradas del índice, luego la opción makeindex genera un archivo con el índice formateado y ordenado alfabéticamente y finalmente al compilar en PDFLaTeX genera el documento PDF con el índice alfabético.

En la figura 11.1 se muestra la diagrama de flujo de la generación de índice.

Figura 11.1



1. En la primera compilación se genera el archivo de extensión `.idx`.
2. El archivo `.idx` junto al archivo `ist` se utilizan como entrada para ser procesados con makeindex y como resultado se obtiene un archivo `.ind` (índice alfabético) y un archivo `.ilg` que contiene el log.
3. Finalmente LaTeX lee el archivo `.ind` dando como resultado la composición tipográfica incluido el índice alfabético

1. Para las pruebas creamos un documento latex de nombre `prueba.tex` con el siguiente contenido:

```
\documentclass{book}
\usepackage{makeidx}
\usepackage[paperwidth=10cm,paperheight=10cm, margin=1cm]{geometry}
\makeindex
\begin{document}
  El comando index crea una entrada para el índice \index{Palabra clave}

  El comando printindex imprime el índice alfabético.
  \printindex
\end{document}
```

```
~/Documents/pruebatex …
+ → ls
.rw-r--r-- 292 userh  6 Nov 14:02 prueba.tex
```

(a)

```
~/Documents/pruebatex …
+ → pdflatex prueba.tex
```

(b)

```
~/Documents/pruebatex …
+ → ls
.rw-r--r--   32 userh  6 Nov 14:14 prueba.aux
.rw-r--r--   30 userh  6 Nov 14:14 prueba.idx
.rw-r--r-- 4.8k userh  6 Nov 14:14 prueba.log
.rw-r--r--  15k userh  6 Nov 14:14 prueba.pdf
.rw-r--r--  292 userh  6 Nov 14:02 prueba.tex
```

(c)

Figura 11.2

En la figura 11.2(a) se muestra el archivo sin haber sido compilado. Ahora procedemos a compilar en pdflatex como como podemos apreciar en la figura 11.2(b). Una vez compilado se obtiene el archivo `.idx`, como se muestra en la figura 11.2(c).

2. Ahora ejecutamos `makeindex prueba.idx` como se muestra en la figura 11.3. Como resultado

```
~/Documents/pruebatex …
+ → makeindex prueba.idx
```

(a)

```
~/Documents/pruebatex …
+ → ls
.rw-r--r--   32 userh  6 Nov 14:14 prueba.aux
.rw-r--r--   30 userh  6 Nov 14:14 prueba.idx
.rw-r--r--  325 userh  6 Nov 14:25 prueba.ilg
.rw-r--r--   59 userh  6 Nov 14:25 prueba.ind
.rw-r--r-- 4.8k userh  6 Nov 14:14 prueba.log
.rw-r--r--  15k userh  6 Nov 14:14 prueba.pdf
.rw-r--r--  292 userh  6 Nov 14:02 prueba.tex
```

(b)

Figura 11.3

obtenemos el archivo `.ind` y el archivo `.ilg` como se muestra en la figura 11.3(b).

3. Finalmente se debe compilar en pdflatex.

Entrada de índice

\index

Para crear una entrada simplemente se debe invocar el comando

```
\index{<key>}
```

Subentradas de índice

El índice admite tres niveles (principal, subentradas y subentradas secundarias), el índice (archivo `.ind`) contiene un entorno de forma lista que admite tres niveles; por ello, para indicar que es un subentrada debemos separar con el caracter `!` del nivel principal.

Ejemplo 11.2

```
\documentclass{book}
\usepackage{makeidx}
\usepackage[paperwidth=10cm,paperheight=10cm, margin=1cm]{geometry}
\makeindex
\begin{document}
  La cadena ``subentrada'' es un subentrada de principal. \index{principal!subentrada}

  La cadena ``subentrada secundaria 1'' es un subentrada secundaria.
  ↪  \index{principal!subentrada!subentrada secundaria 1}
  \printindex
\end{document}
```

1

La cadena “subentrada” es un subentrada de principal.
La cadena “subentrada secundaria 1” es un subentrada secundaria.

Index

2

Rangos de páginas

El comando `\index{...|(}` indica el comienzo de un rango de páginas para la entrada de ídice y el comando `\index{...|)}` indica el final de un rango de páginas.

Ejemplo 11.3

```
\documentclass{book}
\usepackage{makeidx,lipsum}
\usepackage[paperwidth=10cm, paperheight=6.3cm, margin=1mm]{geometry}
\makeindex
\begin{document}
\index{principal|(}\lipsum[1]\index{principal|)}
\printindex
\end{document}
```

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Index

principal, 1

11.2 Introducción a generación de bibliografía y citas

Para realizar referencias a otros trabajos realizados anteriormente por otros autores debemos utilizar referencias bibliográficas. Existe dos formas de realizar estas referencias: utilizando el entorno `thebibliography` (este entorno es proporcionado por LaTeX) y utilizando `bibtex`.

Entorno `thebibliography`

Sintaxis del entorno `thebibliography`:

thebibliography

```
\begin{thebibliography}{99}
  \bibitem{<key>} <text>
\end{thebibliography}
```

En donde `\bibitem{<key>}` es el identificador de la referencia bibliográfica y crear una entrada de la bibliografía, y `<text>` es el texto de la referencia bibliográfica. Para crear una cita bibliográfica se debe recurrir al comando `\cite{<key>}`.

`\bibitem`

`\cite`

Ejemplo 11.4

```
\documentclass{article}
\usepackage[paperwidth=10cm, paperheight=5cm, margin=5mm]{geometry}
\begin{document}
Referencia bibliográfica a \cite{Lamport}, como también podemos citar a \cite{Goossens}.
\begin{thebibliography}{99}
  \bibitem{Goossens} M. Goossens; F, Mittelbach; A. Samarin. {\it The \LaTeX Companion}.
  ↪  Addison-Wesley. 1993.
  \bibitem{Lamport} L. Lamport. {\it \LaTeX}. Addison-Wesley. 1996.
\end{thebibliography}
\end{document}
```

Referencia bibliográfica a [2], como también podemos citar a [1].

References

[1] M. Goossens; F, Mittelbach; A. Samarin. *The LaTeXCompanion*. Addison-Wesley. 1993.

[2] L. Lamport. *LaTeX*. Addison-Wesley. 1996.

Esquema de referencia para bibliografía

Hasta el momento existe 4 métodos para crear referencias bibliográficas: “titulo abreviado”, “autor-año”, “autor-número” y “número”.

1. **Titulo abreviado**. En este método se incrusta directamente el título abreviado en el texto o en pie de página. Ejemplo: "Goossens et al., LGC" o "Goossens, p.23".

2. **Autor-año**. También es conocido como sistema de Harvard y se lista primeramente los nombres de los autores o autor y el año de la publicación. Si un autor ha publicado varios trabajos en un año, entonces se recomienda poner en la bibliografía una letra minúscula después del año (2000a). Las citas de forma autor-año se pueden manejar de manera muy flexible y completa con el paquete natbib. Cuando se tiene un trabajo hecho por varios autores, entonces se debe de escribir el nombre del primer autor y luego se escribe et al.

3. **Autor-número**. En este método se lista primeramente los nombres de los autores o autor y luego un número; generalmente este número esta rodeado por corchetes.

4. **Número**. En este sistema las obras bibliográficas se enumeran consecutivamente. Un texto tiene referencia a estos números y estos números generalmente están dentro de paréntesis o corchetes. Algunas veces en vez de este número suele utilizarse súperíndices numéricos.

Bibliografía con BIBTEX

Para entender el funcionamiento de BIBTEX y LATEX considero que es mejor comenzar con un diagrama de flujo de interacción entre LATEX y BIBTEX. En la figura 11.4 se muestra la diagrama de flujo de interacción entre LATEX y BIBTEX.

1. En la primera compilación se recopilan todos los valores de las citas que aparecen en el documento LATEX y estos son guardados en el archivo de extensión `.aux`.
2. BIBTEX lee el archivo `.aux` generado en el anterior paso y busca estos valores en un base de datos de bibliografías (los bases de datos de bibliografía tienen extensión `.bib`).
3. Con los resultados de la búsqueda de base de datos se genera un entorno completo de `thebibliography` logrando crear el archivo `.bbl`. Las advertencias y errores que devuelve BIBTEX se guarda en el archivo `.blg`.
4. En la segunda compilación de LATEX se lee el archivo `.bbl`.
5. En la tercera compilación de LATEX se resuelven la referencias cruzadas.

El archivo `.bbl` contiene referencias formateadas y `.bst` contiene los estilos. Del análisis se concluye que BIBTEX no lee nuestro documento LATEX y además BIBTEX genera como bibliografía las referencias citadas.

El comando

```
\bibliography{<database1>, <database2>,...,<databasen>}
```

es utilizado para que BIBTEX busque los bases de datos de bibliográficos definidos por el usuario.

`\bibliography`

El comando

```
\bibliographystyle{<style>}
```

es utilizado por BIBTEX para dar formato de bibliografía. Los estilos más conocidos son: `plain`, `apalike`, `alpha`, `abbrv` y `unsrt`

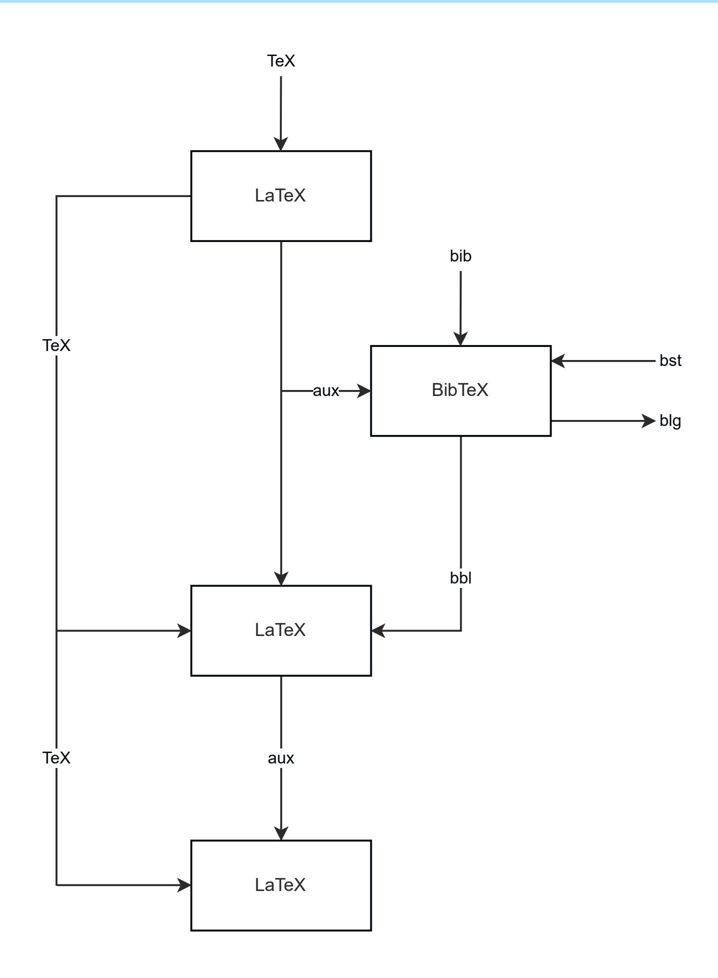
`\bibliographystyle`

Una entrada del archivo `.bib` tiene la siguiente estructura:

```
@type{key,
  property1="value 1",
  property2="value 2",
  ...
}
```

En donde el tipo puede adoptar los valores como

Figura 11.4



1. article
2. book
3. booklet
4. conference
5. inbook
6. incollection
7. inproceedings
8. manual
9. mastersthesis
10. misc
11. other
12. phdthesis
13. proceedings
14. techreport
15. unpublished

y las propiedades pueden adoptar los valores como

1. address
2. abstract
3. author
4. booktitle
5. chapter
6. contents
7. copyright
8. crossref
9. edition
10. editor
11. howpublished
12. institution
13. ISBN
14. ISSN

15 journal
16 keywords
17 language
18 month
19 note
20 number
21 organization
22 pages
23 publisher
24 school
25 series
26 title
27 url
28 volume
29 year

Suponga que se tiene un documento LaTeX con el siguiente contenido:

Ejemplo 11.5

```
\documentclass{article}
\usepackage[paperwidth=10cm,paperheight=5cm,margin=1mm]{geometry}
\begin{document}
Referencia bibliográfica a \cite{Lamport}, como también podemos citar a \cite{Goossens}.
\bibliographystyle{apalike}
\bibliography{dbbiblio}
\end{document}
```

Referencia bibliográfica a [Lamport, 1996], como también pode
mos citar a [Goossens et al., 1993].

References

[Goossens et al., 1993] Goossens, M., Mittelbach, F., and Samarin, A. (1993). *The LaTeX Companion*.

[Lamport, 1996] Lamport, L. (1996). *LaTeX*.

En donde el archivo `bdbiblio.bib` se contiene:

```
@book{Lamport,
  author="Leslie Lamport",
  title="\LaTeX",
  editor="Addison-Wesley",
  year="1996"
}
```

```
@book{Goossens,
  author="Michel Goossens and
  ↪  Frank Mittelbach and
  ↪  Alexander Samarin",
  title="The \LaTeX Companion",
  editor="Addison-Wesley",
  year="1993"
}
```

`\nocite`

Si se desea imprimir alguna entrata sin citar se debe utilizar el comando `\nocite{<key>}`. Para imprimir todas las entradas sin citar se debe invocar el comando `\nocite{*}` en el documento LaTeX.

Ejemplo 11.6

```
\documentclass{article}
\usepackage{lipsum}
\usepackage[paperwidth=10cm, paperheight=10cm, margin=1mm]{geometry}
\begin{document}
\lipsum[1]
\nocite{*}
\bibliographystyle{apalike}
\bibliography{bibliografia}
\end{document}
```

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

References

[A. and F.,] A., A. B. and F., W. M. Edición de textos científicos latex.

[Abrahams et al., 1990] Abrahams, P. W., Hargreaves, K. A., and Berry, K. (1990). *TEX for the impatient*. Addison-Wesley, Reading, Mass., reprinted with corrections edition.

[Goossens, 2010] Goossens, M., editor (2010). *The LATEX graphics companion*. Addison-Wesley series on tools and techniques for computer typesetting. Addison-Wesley, Upper Saddle River, NJ [u.a.], 2. ed., 3. printing edition. Box u.d.T.: The LATEX companions. Rev. set.

[Grätzer, 1996] Grätzer, G. (1996). *Math into LATEX*. Birkhäuser, Boston [u.a.]. Literaturverz. S. [413] - 415.

[Grätzer, 2016] Grätzer, G. (2016). More math into latex.

[Jackson and Goossens,] Jackson, S. and Goossens, L. Handbook of adolescent development.

[Knuth, 2000] Knuth, D. E. (2000). *Computers & typesetting. Volume A–E. Boxed set. Volume A: The TeXbook. Volume B: TeX: the program. Volume C: The Metafontbook. Volume D: Metafont: the program. Volume E: Computer modern typefaces*. Amsterdam : Addison-Wesley Longman, updated Millenium ed. edition.

[Kottwitz, 2011] Kottwitz, S. (2011). Latex beginner's guide. Includes index.

[Lamport, 1995] Lamport, L. (1995). *Das LATEX-Handbuch*. Addison-Wesley, Bonn [u.a.], [2. aufl.] edition. Literaturverz. S. [279].

[Lamport, 1998] Lamport, L. (1998). *LATEX*. Addison-Wesley, Reading, Mass. [u.a.], 2nd ed., 7. + 8. printing edition. Bibliography S. 235.

[Lamport et al., 2022] Lamport, L., Mittelbach, F., and Braams, J. (2022). Standard document classes for latex version 2e. pages 1–58.

[Man, 1992] Man, K. (1992). Tex for the impatient. 14:238.

[Mittelbach and Goossens, 2005] Mittelbach, F. and Goossens, M. (2005). *Der LATEX-Begleiter*. st - scientific tools. Pearson Studium, München [u.a.], 2., überarb. und erw. aufl. edition. Hier auch später erschienene, unveränderte Nachdrucke.

[Mittelbach et al., 2013] Mittelbach, F., Goossens, M., and Braams, J., editors (2013). *The LATEX companion*. Addison-Wesley series on tools and techniques for computer typesetting. Addison-Wesley, Boston, Mass. [u.a.], 2. ed., 12. print. edition. Literaturverz. S. [963] - 982.

[Mojžišová and Pócsová, 2018] Mojžišová, A. and Pócsová, J. (2018). Visualisation of mathematical content using latex animations. pages 536–541, Szilvasvarad, Hungary. IEEE.

[Spivak, 1990] Spivak, M. D. (1990). *The joy of TEX*. American Mathematical Society, Providence, RI, 2. ed edition.

[Syropoulos, 2003] Syropoulos, A. (2003). Digital typography using latex. Includes bibliographical references (p. 469-470) and indexes.

[Tantau, 2007] Tantau, T. (2007). Tutoriel tikz. pages 23–92.

[van Dongen, 2012] van Dongen, M. R. C. (2012). Latex and friends.

Para generar el base de datos de bibliografía existen los programas como JabRef y entre otros.

11.3 Esquema numérico de bibliografía

Comandos estandar LaTeX

Los comandos estándares para referencias en LaTeX son:

```
\cite[text]{<key>}   \cite[<text>]{key1,key2,...}   \nocite{<key>}
```

El comando `\cite` asocia un `key` con el argumento obligatorio de `\bibitem`. El argumento obligatorio de `\cite` es sensible a mayúsculas y minusculas.

Personalización de citas

La personalización de formato de citas se realiza por la redefinición del comando interno de LaTeX. Esta operación es un poco complicado por lo que recomiendo que utilizar algún paquete.

PAQUETES Y CLASES

12.1 Introducción

En este capítulo se abordará a la escritura de clases y paquetes LaTeX, utilizando LaTeX 2.09.

Escribir un paquete y clase para LaTeX 2ε

LaTeX es un sistema de procesamiento de documentos que permite al autor del documento concentrarse en el contenido del mismo, sin preocuparse demasiado por el formateo. Por ejemplo, para indicar un capítulo se utiliza `\chapter{<title>}`, en lugar de seleccionar un tamaño de letra de 18 puntos y en negrita.

El archivo que contiene la información sobre cómo convertir la estructura lógica (como `\chapter`) en formateo (como 18 puntos en negrita y justificado a la izquierda) se llama clase de documento o documentclass. Además, algunas características (como el uso de color o la inclusión de gráficos) son independientes de la clase de documento y se incluyen en paquetes.

Una de las principales diferencias entre LaTeX 2.09 y LaTeX 2ε es en los comandos que se utilizan para escribir paquetes y clases. En LaTeX 2.09, había muy poco soporte para escribir archivos .sty, por lo que los autores tenían que recurrir a utilizar comandos de bajo nivel.

LaTeX 2ε proporciona comandos de alto nivel para estructurar paquetes. También es mucho más fácil construir clases y paquetes unos sobre otros, por ejemplo escribiendo una clase de informe basada en la clase article.

12.2 Escribiendo clases y paquetes

¿Qué es docstrip y doc?

El software doc, que viene con LaTeX, es útil para escribir y documentar grandes fuentes de código en LaTeX. Estas fuentes pueden ser clases o paquetes y pueden ser procesadas de dos maneras: pueden ser ejecutadas a través de LaTeX para producir documentación y pueden ser procesadas con docstrip para producir archivos de clase o paquete. El software doc también puede generar índices automáticamente de definiciones, índices de uso de comandos y listas de cambios. Los archivos .dtx en la distribución de LaTeX son ejemplos de cómo se utiliza doc y docstrip. Puedes obtener más información sobre ellos consultando los archivos docstrip.dtx y doc.dtx

Docstrip es una herramienta diseñada para procesar archivos .dtx (documentado TEX) y extraer de ellos los archivos .cls (clase) y .sty (paquete) necesarios para utilizar una clase o un paquete en LaTeX. Los archivos .dtx contienen tanto el código TEX necesario para implementar la clase o el paquete, como la documentación y ejemplos de uso. Al procesar un archivo .dtx con docstrip, se pueden obtener tres tipos de archivos:

1. Archivos de implementación: contienen el código TEX necesario para implementar la clase o el paquete. Estos archivos suelen tener extensiones .cls o .sty.
2. Archivos de documentación: contienen la documentación de la clase o el paquete. Estos archivos suelen tener extensiones .dvi (archivo DVI), .pdf (archivo PDF) o .ps (archivo PostScript).
3. Archivos de ejemplo: contienen ejemplos de cómo utilizar la clase o el paquete. Estos archivos suelen tener extensiones .tex (archivo TEX) o .dvi (archivo DVI).

Para utilizar docstrip, es necesario incluir comentarios especiales en el archivo .dtx que indiquen qué secciones del archivo deben incluirse en cada uno de estos tres tipos de archivos. Luego, se ejecuta docstrip con el archivo .dtx como entrada y se especifican los archivos de salida deseados. Docstrip procesará el archivo .dtx y extraerá las secciones relevantes a cada archivo de salida.

Es importante destacar que docstrip es solo una herramienta para procesar archivos .dtx y generar archivos.

Identificación de clase y paquete

Cuando quieres añadir nuevos comandos LaTeX a un archivo, es importante decidir si deben ir en una clase de documento o en un paquete. La regla general es: si los comandos se podrían usar con cualquier clase de documento, entonces deben ir en un paquete; en caso contrario, deben ir en una clase.

Existen dos tipos principales de clases: aquellas que son independientes, como article, report o letter, y aquellas que son extensiones o variaciones de otras clases, como la clase de documento proc, que se basa en la clase de documento article.

Por ejemplo, una empresa podría tener su propia clase local de nombre myclass para imprimir cartas. Esta clase se basaría en la clase letter existente, pero no se podría usar con ninguna otra clase de documento, por lo que se tendría myclass.cls en lugar de myclass.sty.

Por otro lado, el paquete graphics proporciona comandos para incluir imágenes en un documento LaTeX. Dado que estos comandos se pueden usar con cualquier clase de documento, por lo que es un archivo de tipo graphics.sty en lugar de graphics.cls.

Nombres de comandos

En LaTeX, hay tres tipos de comandos: los comandos para el autor, como `\section`, `\emph` y `\times`, que tienen nombres cortos y todos en minúsculas; los comandos para los escritores de clases y paquetes, como `\InputIfFileExists`, `\RequirePackage` y `\PassOptionsToClass`, que tienen nombres largos

y mixtos; y los comandos internos utilizados en la implementación de LaTeX, como `\@tempcnta`, `\@ifnextchar` y `\@echa`, que contienen `@` en su nombre y solo deben utilizarse en archivos de clase y paquete. Sin embargo, por razones históricas, a veces es difícil distinguir entre estos comandos. Por ejemplo, `\hbox` es un comando interno que solo se debe utilizar en el núcleo de LaTeX, mientras que `\m@ne` es la constante -1 y podría haber sido algo como `\MinusOne`. Aun así, este regla sigue siendo útil: si un comando tiene `@` en su nombre, entonces no es parte del lenguaje de LaTeX soportado y su comportamiento puede cambiar en futuras versiones como en LaTeX3.

Comandos de caja y colores

La forma más sencilla de asegurar la compatibilidad con LaTeX 2_ε es utilizar los comandos de cajas de LaTeX en lugar de los primitivos de TeX, es decir, utilizar `\sbox` en lugar de `\setbox`, `\mbox` en lugar de `\hbox` y `\parbox` o el entorno minipage en lugar de `\vbox`. Los comandos de cajas de LaTeX ahora tienen opciones nuevas que los hacen tan poderosos como los primitivos de TeX. No siempre se debe cumplir esta regla.

Como ejemplo de lo que puede salir mal, consideremos que en `{\ttfamily <text>}` la fuente se restaura justo antes del }, mientras que en la estructura similar `{\color{blue} <text>}` el color se restaura justo después del último }. Normalmente, esta distinción no importa en absoluto; pero consideremos una asignación de caja primitiva de TeX como:

```
\setbox0=\hbox{\color{blue} <text>}
```

En este caso, la restauración de color se realiza después de } por lo que el color no se almacena en la caja. Este efecto puede causar un serie de problemas en los colores, por ejemplo, se puede generar colores incorrectos en un documento LaTeX hasta puede causar errores en el controlador dvi que se utiliza al momento de imprimir un documento.

El comando `\normalcolor` se utiliza para establecer el color principal del documento en una región de la página. Normalmente, este comando es como `\relax`, que no tiene ningún efecto. Sin embargo, puede utilizar este comando junto con el paquete xcolor para establecer el color principal del documento en una región de la página, como títulos o subtítulos.

Por ejemplo, si ha establecido el color principal del documento en rojo mediante el paquete xcolor, puede utilizar el comando `\normalcolor` para cambiar el color de un título o subtítulo:

Ejemplo 12.1

```
\documentclass{article}
\usepackage{xcolor}
\usepackage{geometry}
\geometry{margin=5mm, paperwidth=10cm, paperheight=7cm}
\begin{document}
{
\color{red}
```

```
\section{Título rojo}
Este es un título escrito en rojo.

\subsection{Subtítulo rojo}
Este es un subtítulo escrito en rojo.

\normalcolor
Este es un párrafo escrito en el color principal del documento (rojo), ya que hemos utilizado el
↪  comando normalcolor para establecer el color de este párrafo en el color principal del documento.
}
\end{document}
```

1 Título rojo

Este es un título escrito en rojo.

1.1 Subtítulo rojo

Este es un subtítulo escrito en rojo.

Este es un párrafo escrito en el color principal del documento (rojo), ya que hemos utilizado el comando normalcolor para establecer el color de este párrafo en el color principal del documento.

Definición caracteres de texto y matemático

Para producir símbolos, acentos y otros elementos en LaTeX, se deben usar comandos especiales. Estos comandos deben definirse de la manera adecuada para que funcionen correctamente, y esto se describe en la documentación de LaTeX llamada “Font Selection” (Selección de fuentes).

Además, es recomendable utilizar el comando `\DeclareRobustCommand` para definir comandos que sean independientes de la codificación. Esto significa que estos comandos funcionarán correctamente independientemente de la codificación que se esté utilizando.

Estilos generales

Cargado de otros archivos

LaTeX proporciana los siguientes comandos:

1. `\LoadClass` es utilizado para cargar una clase de documento.
2. `\LoadClassWithOptions` es utilizado para cargar una clase de documento con opciones.

3. \RequirePackage es utilizado para cargar un paquete.
4. \RequirePackageWithOptions es utilizado para cargar un paquete con opciones.

Se recomienda utilizar estos comandos en lugar del primitivo \input.

En la gran mayoría de situaciones, es recomendable cargar los paquetes utilizando el comando \RequirePackage o \usepackage. Al cargar un paquete varias veces utilizando estos comandos, solo se cargará una vez, lo cual resulta en un ahorro de tiempo y memoria. Por otro lado, si se utiliza el comando \input varias veces, el paquete se cargará repetidamente, lo que puede consumir más memoria y tiempo, además de potencialmente generar resultados inesperados.

No existe la posibilidad de cargar paquetes con opciones por medio del primitivo \input

Si el paquete foo.sty carga el paquete baz.sty mediante el uso de \input, el usuario recibirá una advertencia:

```
LaTeX Warning:  You have requested package ‘foo’,
                but the package provides ‘baz’.
```

Buenas prácticas para escribir paquetes y clases

Cuando se escriben paquetes y clases en LaTeX, es recomendable utilizar los comandos proporcionados por el sistema en la medida de lo posible. Esto se debe a que los comandos de LaTeXhan sido diseñados para evitar problemas comunes y facilitar la creación de archivos robustos y portables. Algunos de los comandos especialmente útiles que conviene tener en cuenta son:

1. \newcommand, \renewcommand y \providecommand: en lugar de usar \def para definir nuevos comandos, es mejor usar uno de estos comandos. Esto ayuda a evitar redefiniciones accidentales de comandos existentes.
2. \newenvironment y \renewenvironment: para definir nuevos entornos de documento, utilice estos comandos en lugar de \def\foo{...} y \def\endfoo{...}.
3. \setlength: para establecer o cambiar el valor de un registro de dimensión o espaciado.
4. \sbox, \mbox y \parbox: estos comandos en vez de \setbox, \hbox y \vbox para manipular cajas.
5. \PackageError, \PackageWarning y \PackageInfo: en lugar de utilizar \@latexerr, @warning y \wlog, para generar mensajes de error, advertencia y información.
6. \DeclareOption y \ProcessOptions: estos comandos utilizan menos memoria que definiendo \ds@<option> y llamando a \@options. Entonces en vez del código

   ```
   \def\ds@draft{\overfullrule 5pt}
   \@options
   ```

   se debe utilizar

```
\DeclareOption{draft}{\setlength{\overfullrule}{5pt}}
\ProcessOptions
```

12.3 Estructura de clase y paquete

Los archivos de clases y paquetes de LaTeX 2_ε tienen una estructura más definida que los archivos de estilo de LaTeX 2.09. Un archivo de clase o paquete se dividiría en los siguientes bloques:

Identificación: Al comienzo del archivo se indica que se trata de un paquete o clase de LaTeX 2_ε y se da una descripción breve de lo que hace.

Declaraciones preliminares: En este bloque se declaran algunos comandos y también se pueden cargar otros archivos. Estos comandos suelen ser solo los necesarios para el código utilizado en las opciones declaradas.

Opciones: Aquí se declaran y procesan las opciones disponibles para el paquete o clase.

Más declaraciones: En este último bloque se hace la mayor parte del trabajo del archivo, como declarar nuevas variables, comandos y fuentes, y cargar otros archivos.

Identificación

Lo primero que hace un archivo de clase o paquete es identificarse a sí mismo. Los archivos de paquetes lo hacen de la siguiente manera:

```
\NeedsTeXFormat{LaTeX2e}
\ProvidesPackage{<package>}[<date> <other information>]
```

Por ejemplo:

```
\NeedsTeXFormat{LaTeX2e}
\ProvidesPackage{latexsym}[2008/01/01 v1.0 Standard LaTeX package]
```

Los archivos de clases lo hacen de la siguiente manera:

```
\NeedsTeXFormat{LaTeX2e}
\ProvidesClass{<class>}[<date> <other information>]
```

Por ejemplo:

```
\NeedsTeXFormat{LaTeX2e}
\ProvidesClass{apart}[2023/01/01 v1.0 APA LaTeX document class]
```

La fecha de lanzamiento se debe dar en el formato “AAAA/MM/DD” y debe estar presente si se utiliza el argumento opcional (esto también es cierto para el comando `\NeedsTeXFormat`). Cualquier derivación

de esta sintaxis resultará en errores de bajo nivel de TeX. Los comandos esperan una sintaxis válida para agilizar el uso diario del paquete o clase y no prevén el caso en el que el desarrollador cometa un error. Esta fecha se verifica cada vez que un usuario especifica una fecha en sus comandos \documentclass o \usepackage. Por ejemplo, si escribió:

```
\documentclass{article}[1995/12/23]
```

los usuarios en otro lugar obtendrían una advertencia de que su copia de article está desactualizada.

La descripción de una clase se muestra cuando se utiliza la clase. La descripción de un paquete se coloca en el archivo de registro. Estas descripciones también se muestran con el comando \listfiles.

Usando clases y paquetes

La primera gran diferencia entre los archivos de estilo de LaTeX 2.09 y los paquetes y clases de LaTeX 2_ε es que LaTeX 2_ε admite la modularidad, en el sentido de construir archivos a partir de pequeños bloques de construcción en lugar de utilizar archivos grandes y únicos.

Un paquete o clase LaTeXpuede cargar otro paquete de la siguiente manera:

```
\RequirePackage{<package>}
```

Por ejemplo:

```
\RequirePackage{ifthen}[1994/06/01]
```

Este comando tiene la misma sintaxis que el comando \usepackage del autor.

Un archivo de clase de LaTeX puede cargar otro archivo de clase de la siguiente manera:

```
\LoadClass[<options>]{<class-name>}[<date>]
```

Por ejemplo:

```
\LoadClass[twocolumn]{article}
```

Este comando tiene el mismo sintaxis que el comando \documentclass para el autor. Permite que las clases se basen en la sintaxis y apariencia de otra clase. Por ejemplo, al cargar la clase article, el autor de una clase solo tiene que cambiar los aspectos de article que no le gusten, en lugar de escribir una nueva clase desde cero.

Los siguientes comandos pueden usarse en el caso común en que solo se desea cargar un archivo de clase o paquete con exactamente las mismas opciones que se están usando en la clase actual:

```
\LoadClassWithOptions{<class-name>}[<date>]
\RequirePackageWithOptions{<package>}[<date>]
```

Por ejemplo:

```
\LoadClassWithOptions{article}
\RequirePackageWithOptions{graphics}{1995/12/01}
```

Declarando opciones

Ahora trataremos sobre cómo se declaran y utilizan las opciones en paquetes y clases de LaTeX. LaTeX es un sistema de composición de texto que se utiliza para producir documentos con un diseño profesional.

Una opción es una forma de especificar un comportamiento o configuración deseado al utilizar un paquete o clase. Por ejemplo, si un autor está escribiendo un documento y quiere que el texto se divida en dos columnas, puede utilizar la opción de `\twocolumn`, que está declarada en la clase de artículo.

Para declarar una opción, se utiliza el comando `\DeclareOption`.

```
\DeclareOption{<option>}{<code>}
```

Este comando tiene dos partes: el nombre de la opción (por ejemplo, "dvips" o "a4paper") y el código que se ejecutará cuando se utilice la opción (por ejemplo, cargar un archivo o establecer la longitud de una variable).

```
\DeclareOption{dvips}{\input{dvips.def}}
```

Cuando un usuario utiliza un paquete o clase con una opción, se ejecuta el código correspondiente a esa opción. Si el usuario utiliza una opción que no ha sido declarada, por defecto se producirá un error o advertencia, dependiendo de si se trata de un paquete o una clase. Sin embargo, se puede cambiar este comportamiento utilizando el comando

```
\DeclareOption*{<option>}{<code>}
```

con el carácter "*", lo que indica que se deben ignorar las opciones no declaradas. Por ejemplo, para que el paquete `prueba.sty` produzca una advertencia en lugar de un error para opciones desconocidos se debe especificar de la siguiente manera:

```
\DeclareOption*{%
  \PackageWarning{prueba}{Unknown option '\CurrentOption'}
}
```

Si se llama al paquete de la forma

```
\usepackage[option1]{prueba}
```

se obtendrá

```
Package prueba Warning: Unknow option 'option1'
```

Los comandos \PassOptionsToPackage y \PassOptionsToClass se utilizan para pasar opciones a otro paquete o clase LaTeX. Por ejemplo, si queremos pasar la opción “color” al paquete “miPaquete”, podríamos utilizar el siguiente código:

```
\PassOptionsToPackage{color}{miPaquete}
```

De esta forma, el paquete “miPaquete” recibiría la opción “color” y podría utilizarla de alguna manera. Por ejemplo, podría utilizarla para cambiar el color de algún elemento en el documento.

El comando \PassOptionsToClass funciona de manera similar, pero en lugar de pasar las opciones a un paquete, las pasa a una clase. Por ejemplo, si queremos pasar todas las opciones desconocidas a la clase “article”, podríamos utilizar el siguiente código:

```
\DeclareOption*{\PassOptionsToClass{\CurrentOption}{article}}
```

Esto le diría a la clase “article” que debe procesar cualquier opción desconocida que se le pase. Es importante asegurarse de cargar la clase “article” en algún momento posterior, ya que de lo contrario las opciones nunca se procesarán. Por ejemplo, si queremos cargar la clase article después de haber pasado las opciones, podríamos utilizar el siguiente código:

```
\usepackage{article}
```

Esta línea de código se debe incluir en algún momento después de haber utilizado el comando \DeclareOption* para pasar las opciones a la clase article. Si no se incluye esta línea de código, las opciones nunca se procesarán y no tendrán ningún efecto en el documento.

Hasta ahora, solo hemos explicado cómo declarar opciones, no cómo ejecutarlas. Para procesar las opciones con las que se llamó el archivo, debe usar:

```
\ProcessOptions\relax
```

Esto ejecuta el <code> para cada opción que se especificó y declaró.Por ejemplo, si el archivo del paquete mipaquete contiene:

```
\DeclareOption{foo}{\typeout{Saw foo.}}
\DeclareOption{baz}{\typeout{Saw baz.}}
\DeclareOption*{\typeout{What's \CurrentOption?}}
\ProcessOptions\relax
```

y un autor escribe

```
\usepackage[foo,bar]{mipaquete}
```

se obtendrá

```
Saw foo.
What's bar?
```

12.4 Ejemplo de clase simple utilizando clase letter

```
\NeedsTeXFormat{LaTeX2e}
\ProvidesClass{myclass}[2023/07/01 Class]
\DeclareOption{onecolumn}{\OptionNotUsed}
\DeclareOption{green}{\renewcommand{\headlinecolor}{\color{green}}
\DeclareOption*{\PassOptionsToClass{\CurrentOption}{letter}}
\ProcessOptions\relax
\LoadClass[a4paper]{letter}
\renewcommand{\ps@firstpage}{%
  \renewcommand{\@oddhead}{⟨letterhead goes here⟩}%
  \renewcommand{\@oddfoot}{⟨letterfoot goes here⟩}%
}
```

DOCUMENTCLASS

13.1 Introducción

En este capítulo, se va a mostrar cómo crear una plantilla para un libro utilizando diferentes paquetes. En un próximo capítulo, se explicará cómo crear plantillas utilizando los comandos definidos en el núcleo de LaTeX.

En este libro, se utilizará el término "plantilla" para referirse a un "documentclass".

Se requieren ciertos conocimientos para seguir este capítulo, entre ellos:

1. Tikz: se utilizará para crear diferentes formatos o cajas, como la enumeración de páginas, entre otros.
2. Tcolorbox: se utilizará para crear entornos de teoremas, como definiciones, corolarios, ejemplos de solución de problemas, entre otros.
3. Introducción a los macros de TeX: es necesario tener una breve introducción sobre los macros de TeX, como por ejemplo, conocer el macro `\def` de TeX.
4. Todos los capítulos avanzados previos a este, excepto el capítulo de asymptote.

13.2 ¿En qué orden crear el diseño del libro?

Desde mi punto de vista, el orden adecuado para diseñar el formato de un libro sería el siguiente:

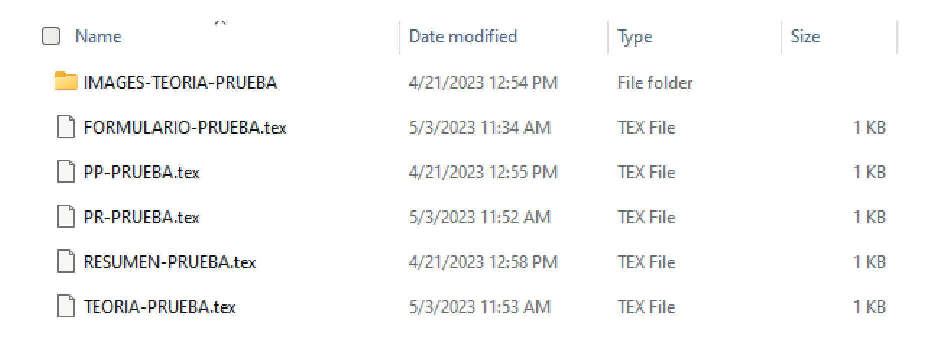
1. Establecer la estructura del libro.
2. Definir la estructura de la clase.
3. Crear un macro que administre la estructura del libro.
4. Definir las dimensiones geométricas del documento.
5. Otros aspectos a considerar.

Estructura del libro

La gestión de documentos extensos, como los libros, puede presentar un desafío en cuanto a su organización. Es fundamental conocer la forma adecuada de estructurar el libro y organizar los capítulos y secciones para que su navegación y lectura sean sencillas.

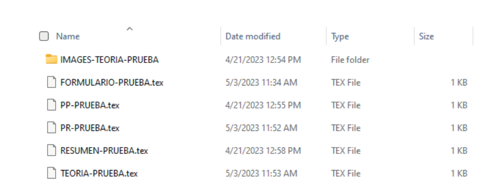
Otro aspecto importante en la gestión de documentos extensos es el tiempo de compilación. A medida que el documento crece en tamaño, el tiempo de compilación también aumenta significativamente. Para minimizar este tiempo, se puede crear un formato personalizado.

Figura 13.1



Name	Date modified	Type	Size
IMAGES-TEORIA-PRUEBA	4/21/2023 12:54 PM	File folder	
FORMULARIO-PRUEBA.tex	5/3/2023 11:34 AM	TEX File	1 KB
PP-PRUEBA.tex	4/21/2023 12:55 PM	TEX File	1 KB
PR-PRUEBA.tex	5/3/2023 11:52 AM	TEX File	1 KB
RESUMEN-PRUEBA.tex	4/21/2023 12:58 PM	TEX File	1 KB
TEORIA-PRUEBA.tex	5/3/2023 11:53 AM	TEX File	1 KB

En la figura 13.1 se muestra la estructura del libro, la cual consta de tres archivos principales (main, colors y artbook) y tres carpetas (PRUEBA, images y imagesCapitulo). El archivo main es cualquier documento LaTeX que contiene el entorno document. El archivo colors contiene las definiciones de colores que se utilizarán en el libro y se puede tratar como un paquete, ya que su extensión es sty. El archivo artbook contiene las definiciones de formato de una clase. En la carpeta images se almacenan las imágenes del libro, como la portada, foto del autor y entre otros. Los capítulos del libro se almacenan en carpetas, donde el nombre de la carpeta corresponde al nombre del capítulo (por ejemplo, en este caso es PRUEBA).



Name	Date modified	Type	Size
IMAGES-TEORIA-PRUEBA	4/21/2023 12:54 PM	File folder	
FORMULARIO-PRUEBA.tex	5/3/2023 11:34 AM	TEX File	1 KB
PP-PRUEBA.tex	4/21/2023 12:55 PM	TEX File	1 KB
PR-PRUEBA.tex	5/3/2023 11:52 AM	TEX File	1 KB
RESUMEN-PRUEBA.tex	4/21/2023 12:58 PM	TEX File	1 KB
TEORIA-PRUEBA.tex	5/3/2023 11:53 AM	TEX File	1 KB

Figura 13.2

Dentro de cada carpeta de capítulo se encuentra una subcarpeta llamada "IMAGE-TEORIA-..." y cinco archivos con extensión .tex. El archivo "IMAGE-TEORIA-PRUEBA" contiene las imágenes que se utilizarán en el capítulo. El archivo "PP-..." incluye los problemas propuestos del capítulo, el archivo "PR-..." contiene los problemas resueltos, el archivo "RESUMEN-..." contiene el resumen y el archivo "TEORIA-..." incluye la parte teórica del capítulo, como se muestra en la figura 13.2.

Estructura de clase

Es importante establecer la estructura de una clase para evitar posibles conflictos entre los distintos paquetes que se utilizan al diseñar una plantilla. Dado que en un libro se emplean múltiples paquetes, estos pueden entrar en conflicto entre sí.

La estructura de la clase para esta plantilla será similar a la estructura de la clase estándar "book". En la sección 13.5, se detalla minuciosamente la estructura de esta plantilla.

Macro gestionante de la estrutura del libro

Una vez que se ha estructurado la clase, es necesario crear un macro que administre la estructura del libro, es decir, un macro que importe los archivos correspondientes de los capítulos del libro. Este macro se denomina `\chapterfile`, y es el encargado de gestionar la estructura del libro.

Dimensiones geométricas del documento LATEX

Las dimensiones del libro, como el ancho y la altura, se definen mediante el uso del paquete `geometry`, el cual permite establecer diferentes tipos de páginas con geometrías distintas.

Otros

Una vez se ha estructurado el libro, definido la estructura de la clase y creado un macro que administre la estructura del libro, se puede dar formato a otros elementos como listas, notas flotantes en el margen, secciones, estilos de página, entre otros. En mi opinión, lo recomendable es realizar todo esto siguiendo la estructura de la clase previamente definida.

13.3 Introducción a macros TEX

LATEX está construido sobre macros de bajo nivel denominados macros TeX. El término comando es introducido en LaTeX y mientras tanto el término macro es utilizado en TeX.

Macro def

En TeX, los macros son secuencias de comandos que se pueden definir para realizar tareas específicas y automatizar procesos repetitivos. El comando de TeX para definir un macro es `\def`.

Sintaxis del macro def:

```
\def\nombremacro...#1#2...#9{...}
```

En donde:

1. `\def` es el comando para definir un macro.
2. `\nombremacro` es el nombre del macro.
3. `#1` es el primer parámetro del macro.
4. `#2` es el segundo parámetro del macro.

5 #9 es el noveno parámetro del macro.

6 . . . es el cuerpo del macro.

A continuación, se presentan algunos ejemplos que ilustran cómo se puede utilizar `\def` en TeX.

Definición de macros simples

El siguiente ejemplo define un macro simple que coloca un signo de exclamación después de su argumento.

Ejemplo 13.1

```
\def\teopit{Teorema de Pitágoras: $c^2 = a^2 + b^2$}
\teopit
```

Teorema de Pitágoras: $c^2 = a^2 + b^2$

Definición de macros con parámetros

En TeX, los macros con parámetros obligatorios son aquellos que toman uno o varios argumentos que son necesarios para que el macro funcione correctamente. Para definir un macro con parámetros obligatorios, se utiliza el comando `\def` seguido del nombre del macro y la lista de parámetros espesificados por `#1, #2,...`.

Definición de un macro que llama otro macro

En TeX, también se pueden definir macros que llamen a otros macros. El siguiente ejemplo define un macro que llama al macro `\exclam{}`:

Ejemplo 13.2

```
\def\exclam#1{#1!}
\def\greet#1{Hola, #1!\exclam{Gracias por venir a mi
↪  fiesta}}
\greet{Juan}
```

Hola, Juan!Gracias por venir a mi fiesta!

Con esta definición, se puede usar el comando `\greet{}` para saludar a alguien y agradecerle por venir a una fiesta.

Ejemplo 13.3

```
\def\ecuacion#1#2{\ensuremath{#1_1^2+#1_2^2+\cdots
↪  +#1_#2^2=1}}
\ecuacion{z}{5}
```

$z_1^2 + z_2^2 + \cdots + z_5^2 = 1$

El comando `\def`, a diferencia del `\newcommand`, permite elegir los delimitadores de los argumentos.

Ejemplo 13.4

```
\def\ecuacion#1;#2:{\ensuremath{#1_1^2+#1_2^2+\cdots
 ↪  +#1_#2^2=1}}
\ecuacion z;5:
```

$z_1^2+z_2^2+\cdots+z_5^2=1$

Cuando se quiere definir un comando con argumentos opcionales se recomienda acudir al comando `\newcommand`.

Definiciones globales

El alcance de un macro definido por medio del comando def puede tener un alcance local. Esta localidad o scope puede ser un entorno o comando; por ejemplo, si se define un macro por medio del comando def dentro de un entorno o algún comando o macro, la definición del macro solo se expandirá dentro del macro o entorno.

Supongamos que definimos un macro de nombre mymacro:

```
\def\mymacro{Macro fuera del entorno o comando}
```

Ahora vamos a definir otro macro de mismo nombre dentro de la definición de un entorno.

```
\newenvironment{myenv}{Inicio de entorno \par \def\mymacro{Macro dentro del
 ↪  entorno}}{\par Fin de entorno}
```

Cuando invoquemos el macro de nombre mymacro dentro del entorno myenv expandirá o imprime “Macro dentro del entorno”.

Ejemplo 13.5

```
\def\mymacro{Macro fuera del entorno o comando}
\newenvironment{myenv}{%
  Inicio de entorno \par
  \def\mymacro{Macro dentro del entorno}
}{%
  \par Fin de entorno
}
\mymacro

Llamando dentro del entorno:\\
\begin{myenv}
  \mymacro
\end{myenv}

\mymacro
```

Macro fuera del entorno o comando
Llamando dentro del entorno:
Inicio de entorno
Macro dentro del entorno
Fin de entorno
Macro fuera del entorno o comando

Si deseamos que el macro `\mymacro` definido dentro del entorno tenga alcance global (también fuera del entorno) debemos utilizar la macro `\gdef`. El macro `\gdef` es de mismo sintaxis que el macro `\def` solo que define globalmente al macro.

Ejemplo 13.6

```
\def\mymacro{Macro fuera del entorno o comando}
\newenvironment{myenv}{%
  Inicio de entorno \par
  \gdef\mymacro{Macro dentro del entorno}
}{%
  \par Fin de entorno
}
\mymacro

Llamando dentro del entorno:\\
\begin{myenv}
  \mymacro
\end{myenv}

\mymacro
```

Macro fuera del entorno o comando
Llamando dentro del entorno:
Inicio de entorno
Macro dentro del entorno
Fin de entorno
Macro dentro del entorno

Para definir una macro globalmente debemos utilizar una de las dos alternativas:

```
\global\def\NuevoMacro#1...#9{Definición}
\gdef\NuevoMacro#1...#9{Definición}
```

Macro edef

El macro edef, a diferencia del macro def expande todos los macros dentro de su definición.

Ejemplo 13.7

```
\def\a{Hola}
\edef\b{\a, mundo}
\b
```

Hola, mundo

Aplicación de edef en kernel de LaTeX:

```
\DeclareRobustCommand\fontfamily[1]{\edef\f@family{#1}}
```

Macro let

El macro `\let` realiza una copia de un macro ya definido. Sintaxis del macro `\let`:

```
\let\NuevoComando=\ComandoExistente
```

Ejemplo 13.8

```
\def\mymacroa{Hola mundo}
\let\mymacrob=\mymacroa
\mymacrob
```

Hola mundo

Registros

Contador, un forma de registro

TEX proporciona varias variables diferentes y registros asociados que se pueden manipular libremente. Uno de ellos es el registro `\count`⟨num⟩, que es un registro de enteros que proporciona aritmética de enteros de 32 bits.

Existen 256 registros enteros en total, numerados del 0 al 255, y se pueden usar para asignar valores numéricos, como se muestra en los siguientes ejemplos:

```
\count0=42 % asigna el valor 42 al registro 0
\count7=\macro % asigna el valor de \macro al registro 7
```

También se puede obtener el valor de un registro utilizando el comando `\the`⟨registro⟩, como se muestra a continuación:

```
\the\count0 % muestra el valor del registro 0, que es 42
```

Los registros también se pueden usar con macros. Por ejemplo:

```
\def\macro{-123456}
\count0=\macro % asigna el valor de \macro al registro 0
\the\count0 % muestra el valor del registro 0, que es -123456
```

También es posible asignar varios valores a un registro utilizando varias macros:

```
\def\firstmacro{123}
\def\secondmacro{456}
\def\thirdmacro{789}
\count0=\firstmacro\secondmacro\thirdmacro % asigna 123456789 al registro 0
\the\count0 % muestra el valor del registro 0, que es 123456789
```

Sin embargo, se debe tener cuidado al manipular macros y registros, ya que TEX expandirá todo lo que sigue en una asignación, incluso si hay varias macros juntas. En algunos casos, esto puede dar resultados inesperados. Para evitar esto, se puede usar el comando `\relax` después de una asignación para detener el proceso de escaneo, como se muestra en el siguiente ejemplo:

```
\def\macro{1234}
\count0=\macro\relax % se detiene el proceso de escaneo
\the\count0 % muestra el valor del registro 0, que es 1234
```

El macro `\newcount` de LaTeX se define como un `\count` de TeX.

Dimensiones

Una longitud de LaTeX es una forma de dimensión de TeX. Las longitudes se pueden usar para especificar distancias, espacios, tamaños de cajas, etc. Las longitudes se pueden especificar en cualquier unidad de medida, como puntos, pulgadas, centímetros, milímetros, etc. Las longitudes se pueden especificar de varias maneras, como se muestra a continuación:

La instrucción `\dimen`⟨num⟩ de TeX define un registro para almacenar números de punto fijo con unidad, es decir, valores que representan una longitud o dimensión física. Estos registros funcionan de manera similar a los registros `\count`, pero las asignaciones de los registros `\dimen` requieren una unidad explícita, como "cm" o "pt".

Al igual que con los registros `\count`, se puede acceder al valor almacenado en un registro `\dimen` utilizando la instrucción `\the`⟨register⟩.

Ejemplo 13.9

```
\dimen0=1pt
Valor de dimen0: \the\dimen0.
```

Valor de dimen0: 1.0pt.

Ejemplo 13.10

```
\def\macro{1234.5678}
\dimen0=\macro pt
Valor de dimen0: \the\dimen0.
```

Valor de dimen0: 1234.5678pt.

Es importante tener en cuenta que se aplican las mismas reglas de expansión de macros después de las asignaciones de los registros `\dimen`, por lo que se debe tener cuidado para evitar resultados inesperados.

En LaTeX, el macro `\newdimen` se define usando `\dimen` de TeX.

Tokens

Las token registers se pueden pensar como variables de cadena especiales que pueden contener tokens de cualquier tipo, incluyendo caracteres especiales como "#" que normalmente tendrían un significado especial dentro de las macros.

A diferencia de las variables de cadena creadas con `\def\macro{contenido}`, el contenido de las token registers no se expande automáticamente cuando se usan con `\the\toks<number>`. Esto permite un control más fino sobre la expansión, lo que puede ser útil en ciertos casos.

Por ejemplo:

```
\toks0={abc}%
\toks1={DEF}%
```

El valor de `\toks0` es ahora "abc" y el valor de `\toks1` es "DEF". Si se usa `\the\toks0` y `\the\toks1` en el siguiente código, el resultado será "abcDEF":

```
\the\toks0 \the\toks1.
```

También es posible usar el token "#" en una token register:

```
\toks0={#1}%
\message{Meaning is \the\toks0}%
```

Esto producirá un mensaje en el archivo de registro que dice "Meaning is ##1". El doble "#" se debe a que dentro de un mensaje, el token "#" se interpreta como una instrucción especial para imprimir argumentos de macros.

Las token registers son particularmente útiles para el control detallado de la expansión y se discuten con más detalle en estructura de procesador TeX.

Skip

Es conocido en LaTeX como una longitud elástica. En LaTeX se define el comando `\newskip` como un `\skip` de TeX.

Muskip

Es conocido en LaTeX como una longitud elástica en modo matemático. En LaTeX se define el comando `\newmuskip` como un `\muskip` de TeX.

Asignación de registros

Hay un número muy limitado de registros. En consecuencia, uno tiene que pensar cuidadosamente cómo asignarlos. Los casos de uso típicos para los registros son variables temporales (como algún resultado intermedio) y recursos de larga vida que deben acumularse mientras se genera el documento o alguna parte de él. Es claramente una mala idea sobrescribir un registro descuidadamente por lo que se debe tener cuidado en sobreescribir.

TEX viene con una única forma de "asignar" registros:

```
\newdimen⟨\macroname⟩
\newcount⟨\macroname⟩
\newtoks⟨\macroname⟩
```

Estas macros asignan un nuevo registro que luego es accesible como `\macroname`.

Ejemplo 13.11

```
\newdimen\variable
\variable=42pt
El valor es: \the\variable.
```

El valor es: 42.0pt.

El resultado de `\macroname` puede ser utilizado de la misma manera que si se utilizara el registro directamente. De hecho, a menudo es más simple porque no es necesario preocuparse por el número del registro. La asignación se realiza mediante una variable entera global que se incrementa cada vez que se asigna un registro. Esto garantiza que las variables almacenadas en estos registros asignados no sobrescriben accidentalmente el contenido de otra variable. Es importante tener en cuenta que la desasignación no es posible. Es razonable asignar variables de larga duración, pero se debe evitar la asignación de una nueva variable solo porque se necesita una variable temporal “nueva”. Es mejor asignar un par de variables con nombre como `\tempa`, `\tempb` (en el kernel de LaTeX esta práctica se realiza bastante), o algo similar y reutilizar estos valores para cada evaluación temporal. Se debe tener cuidado para evitar sobrescribir involuntariamente. También es posible utilizar registros de tokens, tal como se explicó anteriormente. Sin embargo, se debe proteger su uso mediante grupos:

Ejemplo 13.12

```
\toks3={Valor fuera del grupo}

\begingroup
\toks3={Valor dentro del grupo}
toks3 dentro del grupo: \the\toks3
\endgroup

toks3 fuera del grupo: \the\toks3
```

toks3 dentro del grupo: Valor dentro del grupo
toks3 fuera del grupo: Valor fuera del grupo

Los grupos son el concepto de “ámbito o scope” en TeX.

Operaciones con registros

```
\advance⟨register⟩ by⟨quantity⟩
```

Ejemplo 13.13

`\count0=42`
`\advance\count0 by 10`
`\the\count0`
	52

Ejemplo 13.14

`\dimen0=1pt`
`\advance\dimen0 by 10pt`
`\the\dimen0`
	11.0pt

```
\multiply⟨register⟩ by⟨integer⟩
```

Ejemplo 13.15

`\count0=42`
`\multiply\count0 by -10`
`\the\count0.`
	-420.

Ejemplo 13.16

`\dimen0=0.5pt`
`\multiply\dimen0 by 20`
`\the\dimen0.`
	10.0pt.

```
\divide⟨register⟩ by⟨integer⟩
```

Ejemplo 13.17

`\count0=5`
`\divide\count0 by 2`
`\the\count0`
	2

Ejemplo 13.18

`\dimen0=10pt`
`\divide\dimen0 by 20`
`\the\dimen0`
	0.5pt

```
\dimen⟨number⟩=⟨fixed point number without unit⟩\dimen⟨number⟩
```

Ejemplo 13.19

```
\dimen1=50pt
\dimen0=0.6\dimen1
\the\dimen0.
```

30.0003pt.

En vez de `\count0` y `\dimen0` también podemos utilizar macros.

Herramientas para depurar el código

```
\message{⟨tokens⟩}
\meaning⟨\macro⟩
\tracingmacros=2
\tracingcommands=2
\tracingrestores=1
```

Alcance de un variable

Cada lenguaje de programación conoce el concepto de ámbito: limitan el efecto de las variables o rutinas. Sin embargo, los mecanismos de ámbito de TeX no han sido diseñados para la programación, sino para la composición tipográfica. En TEX, los grupos delimitados por llaves se utilizan para definir el ámbito. Al igual que en otros lenguajes de programación, TEX destruirá cualquier variable (macro) definida dentro de un ámbito al final del mismo, pero también deshará cualquier cambio que se haya aplicado dentro de ese ámbito. TEX no permite escribir en macros de un ámbito externo, excepto en el ámbito superior (global).

TEX tiene tres formas de crear grupos: mediante llaves, `\begingroup` y `\bgroup`. El uso de llaves es menos flexible que las otras dos opciones y debe haber un número igual de llaves abiertas y cerradas. `\begingroup` y `\endgroup` crean y cierran un ámbito respectivamente, y pueden estar separados por otros comandos. `\bgroup` y `\egroup` también crean y cierran un ámbito, pero son más útiles para delimitar argumentos y para probar si el siguiente carácter es una llave de apertura.

Cajas TeX

El sistema TeX al momento de procesar un documento, compone el texti en forma de cajas de 3 dimensiones, altura, anchura y profundidad que se miden con respecto a un punto de referencia. Por ejemplo, para componer un párrafo se crea una lista horizontal y luego está lista se rompe en tamaños determinados, y estas cajas se añaden en una caja vertical.

Modos de TeX

El sistema TEC maneja 3 modos de trabajo: horizontal, vertical y matemático.

En el modo horizontal las cajas son agrupadas horizontalmente juntas unos y otras a lo largo de una línea base. La caja en donde se contiene estas cajas es de anchura igual a la suma de las anchuras de las cajas,

y la altura y profundidad de la caja se adopta el mayor valor de las cajas dentro de ella. Existen dos sub modos diferentes dentro del modo horizontal: el modo ordinario y el modo restringido. El modo ordinario se utiliza cuando se construye párrafos. En el modo restringido se alinean las cajas horizontalmente sin la posibilidad de dividirse la caja resultante en cajas más pequeñas.

En el modo vertical se apilan las cajas verticalmente unas sobre otras, manteniendo su punto de referencia en una misma línea vertical. El ancho de la caja es igual al ancho de las cajas de mayor valor y la altura total es la suma de las alturas de las cajas y profundidades. También existen dos sub modos: el modo ordinario y el modo interno. En el modo ordinario las cajas se empaquetan verticalmente con la posibilidad de someterse. En el modo interno se crea una caja vertical indivisible.

El modo matemático es utilizado para escribir fórmulas y símbolos matemáticos.

Las cajas esperados por sistema TeX.

- `\hbox`<box specification><horizontal material>
- `\vbox`<box specification><vertical material>
- `\vtop`<box specification><vertical material>
- `\box`<8-bit number>
- `\copy`<8-bit number>
- `\vsplit`<8-bit number>to<dimen>
- `\lastbox`

Una `<box specification>` se define como:

```
<box specification> → <filler>
| to <dimen><filler> | spread <dimen><filler>
```

Un `<8-bit number>` es un número en el rango 0-255.

Cajas horizontales

Con fines didácticos recurriremos al comando fbox de LaTeX para dibujar la caja.

Para crear cajas horizontales se utiliza el comando `\hbox{Material}`. El material de la caja se procesa en modo horizontal restringido.

El comando:

```
\hbox to Ancho{Material}
```

Crea una caja horizontal de ancho `Ancho`.

Ejemplo 13.20

```
\parindent 0pt%
Sin definir ancho de la caja: \fbox{\hbox{contenido de la caja}}. Después de la caja%

Definiendo ancho de la caja: \fbox{\hbox to 5cm {contenido de la caja}}. Después de la caja%

Definiendo ancho de la caja: \fbox{\hbox to 1cm {contenido de la caja}}. Después de la caja%
```

Sin definir ancho de la caja: contenido de la caja. Después de la caja
Definiendo ancho de la caja: contenido de la caja. Después de la caja
Definiendo ancho de la caja: contenido de la caja. Después de la caja

El comando:

```
\hbox spread Ancho{Material}
```

Crea una caja horizontal de ancho aumentado o reducido, dependiendo del signo, del valor del ancho natural de la caja.

Ejemplo 13.21

```
\parindent 0pt
Estirado: \fbox{\hbox spread 5mm{contenido de la caja}}. Después de la caja.

Contraído: \fbox{\hbox spread -5mm{contenido de la caja}}. Después de la caja.
```

Estirado: contenido de la caja. Después de la caja.
Contraído: contenido de la caja. Después de la caja.

Algunas aplicaciones de cajas horizontales:

```
\def\leftline#1{\hbox to \hsize{#1\hss}}
\def\rightline#1{\hbox to \hsize{\hss#1}}
\def\centerline#1{\hbox to \hsize{\hss#1\hss}}
\def\rlap#1{\hbox to 0pt{#1\hss}}
\def\llap#1{\hbox to 0pt{\hss#1}}
```

Cajas verticales

Al igual que las cajas horizontales, las cajas verticales se construyen con el comando `\vbox{Material}`. Este comando inicia el modo vertical interno. El ancho de una caja vertical está dado por el `\hsize` y este comando puede estar definido antes de invocar el comando de `\vbox` o como argumento.

Ejemplo 13.22

```
\parindent 0pt%
\hsize\dimexpr0.8\linewidth\relax
\fbox{%
  \vbox{Texto de prueba \hbox{Una caja} \hbox{Otracaja}}
}
\fbox{
  \vbox{\hbox{Una caja} Texto de prueba \hbox{Otracaja}}
}
\fbox{
  \vbox{\hbox{Una caja} \hbox{Otra caja} \hbox{Otra caja mas}}
}
\fbox{
  \vbox{\hsize 4cm Texto de prueba \hbox{Una caja} \hbox{Otra caja}}
}\par
% diferencia entre modos horizontal y vertical:
\fbox{\vbox{Texto de prueba}} \par
\fbox{\vbox{\hbox{Texto de prueba}}}
```

Texto de prueba Una caja Otracaja

Una caja
Texto de prueba Otracaja

Una caja
Otra caja
Otra caja mas

Texto de prueba Una caja
Otra caja

Texto de prueba

Texto de prueba

Al igual que con cajas horizontales, una caja vertical se puede definir su altura

```
\vbox to Alto{Material}
\vbox spread Alto{Material}
```

Existen también los comandos:

```
\vtop to Ancho{Material}
\vcenter to Ancho{Material}
```

que crean cajas verticales alineadas en la parte superior y en el centro, respectivamente.

Ejemplo 13.23

```
\parindent 0pt
\fbox{\vbox to 5mm{\hbox{Texto 1}}}
\fbox{
  \vbox to 10mm{\hsize 3cm%
      \vfil\centerline{Texto 2}\vfil}
}
\fbox{
  \vtop to 5mm%
    {\vfil\hbox{Texto 3}}
}
```

Texto 1 Texto 2 Texto 3

Moviendo cajas

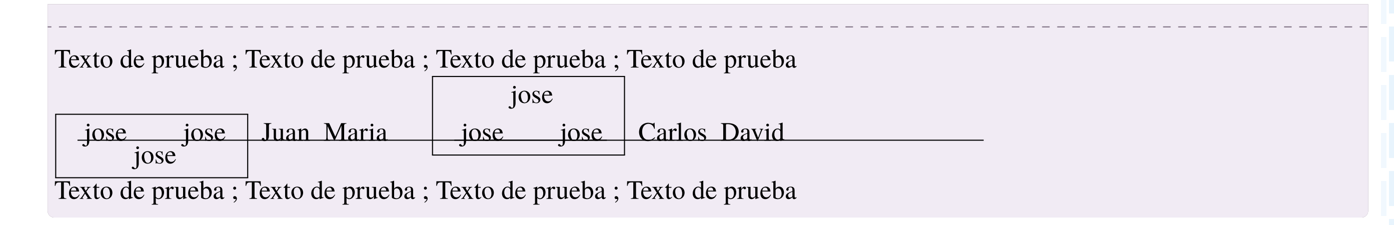
Cuando nos encontremos en modo horizontal, las cajas se pueden desplazar verticalmente. Para ello se utilizan los comandos:

```
\raise<length>{<box>}
\lower<length>{<box>}
```

Este es los comandos son similares y la operación `\raise D = \lower - D` es verídica, y este es comandos no alteran la línea base. La línea base podemos de marcarlo con el comando `\hrule`.

Ejemplo 13.24

```
Texto de prueba ; Texto de prueba ; Texto de prueba ; Texto de prueba \\
\fbox{
  \hbox{
    \hbox to 0pt{\vbox{\hrule width 7cm}} \hbox{jose}
    \lower3mm\hbox{jose} \hbox{jose}
  }
}
\ Juan \ Maria\hspace{5mm}
\fbox{
  \hbox{
    \hbox to 0pt{\vbox{\hrule width 7cm}} \hbox{jose}
    \raise5mm\hbox{jose} \hbox{jose}
  }
}
\ Carlos \ David\\
Texto de prueba ; Texto de prueba ; Texto de prueba ; Texto de prueba
```



Para mover horizontalmente los componentes de una caja vertical, se utilizan los comandos:

```
\moveleft<length>{<box>}
```

Ejemplo 13.25

```
\frame{
  \vbox{
    \hbox{\frame{T}}%
    \hbox{\frame{E}}
    \hbox{\frame{X}}
  }
}
\hspace{1cm}
\frame{
  \vbox{
    \moveleft10pt\hbox{\frame{T}}%
    \moveright10pt\hbox{\frame{E}}%
    \hbox{\frame{X}}%
  }
}%
\hspace{1cm}
\frame{
  \vbox{
    \hbox{\frame{T}}%
    \moveleft12pt\hbox{\frame{E}}%
    \moveright12pt\hbox{\frame{X}}
  }
}
```

T E X   T E X   T E X

Operaciones con cajas

Creando una nueva caja:

```
\newbox\nombrecaja
```

A una caja creada con `\newbox` se le puede asignar un contenido con:

```
\setbox\nombrecaja{<box>}
```

Ejemplos de asignación de contenido a una caja:

```
\setbox\cajaa = \hbox{contenido}
\setbox\cajab = \hbox to 3cm{contenido}
\setbox\cajac\hbox{contenido}
\setbox\cajad\hbox to 3cm{contenido}
```

Las asignaciones de cajas verticales se realizan de manera similar.

En vez de declarar una caja, también podemos utilizar registros de cajas:

```
\setbox<register> = <box>
```

Ejemplos de asignación de contenido a una caja:

```
\setbox1 = \hbox{contenido}
\setbox2 = \hbox to 3cm{contenido}
\setbox3\hbox{contenido}
\setbox4\hbox to 3cm{contenido}
```

Para utilizar una caja y luego borrarlo de la memoria, se utiliza el comando:

```
\box\nombrecaja
```

Para utilizar una caja sin borrarlo de la memoria se debe utilizar el comando:

```
\copy\nombrecaja
```

En el kernel de LaTeX se define una caja temporal que se puede utilizar para almacenar temporalmente un contenido. Esta caja se llama `\@tempboxa`.

Ejemplo 13.26

```
\makeatletter
\newbox\@mytempbox
\setbox\@mytempbox\hbox{
  \begin{tabular}{lcr}
    {\bfseries Nombre} & {\bfseries Apellido} & {\bfseries Edad} \\ \midrule
    Juan               & Pérez                & 20               \\
    María              & González             & 21               \\
    Carlos             & López                & 22               \\
    David              & Martínez             & 23
```

```
  \end{tabular}
}
\begin{center}
  \begin{tikzpicture}
    \node[fill=Salmon!50!white]{\box\@mytempbox};
  \end{tikzpicture}
\end{center}
\makeatother
```

Nombre	Apellido	Edad
Juan	Pérez	20
María	González	21
Carlos	López	22
David	Martínez	23

Cuando se crea una caja, sus dimensiones como la anchura, la altura y la profundidad de la caja creada, se pueden recuperar con los comandos:

```
\wd\nombrecaja
\ht\nombrecaja
\dp\nombrecaja
```

Estos valores se pueden almacenar en dimensiones TeX y en longitudes de LaTeX.

Sistematización de tareas

En esta sección automatizaremos a que un serie de comandos o tareas se ejecuten antes de iniciar un párrafo, fórmula, caja y entre otros. Para este objetivo existen las posibilidades de usar los comandos:

1. `\everypar{<comandos>}`. Se ejecuta el argumento obligatorio, en este caso los comandos, antes de dar formato a cada párrafo.

2. `\everymath{<comandos>}`. Se ejecuta el argumento obligatorio, en este caso los comandos, antes de entrar a modo matemático.ordinario o modo texto.

3. `\everydisplay{<comandos>}`. Se ejecuta el argumento obligatorio, en este caso los comandos, antes de entrar a modo matemático resaltado.

4. `\everyhbox{<comandos>}` y `\everyvbox{<comandos>}`. Se ejecuta el argumento obligatorio, en este caso los comandos, antes de crear una caja horizontal o vertical.

Ejemplo 13.27

```
\parindent 0pt \parskip 10pt
\newcounter{parrafo}
\setcounter{parrafo}{0}
\everypar{
  \addtocounter{parrafo}{1}
  \centerline{\bfseries Párrafo \theparrafo}\\[1mm]
}

Primer párrafo.

Segundo párrafo.

Tercer párrafo.
```

Párrafo 1

Primer párrafo.

Párrafo 2

Segundo párrafo.

Párrafo 3

Tercer párrafo.

Condicionales, bucles y variables booleanas

Condicionales

La construcción `\ifnum` compara números enteros o registros de números enteros (`\count registers`) y contiene dos ramas, una que se ejecuta en el caso verdadero y otra en el caso falso. Sintaxis de `\ifnum`:

```
\ifnum⟨expression⟩⟨comparison⟩⟨expression⟩
  ⟨true code⟩
\else
  ⟨false code⟩
\fi
```

La construcción `\ifdim` es similar a `\ifnum`, pero compara dos números de punto fijo o registros de `\dimen`. Los números deben tener una unidad. Sintaxis de `\ifdim`:

```
\ifdim⟨expression⟩⟨comparison⟩⟨expression⟩
  ⟨true code⟩
\else
```

```
  ⟨false code⟩
\fi
```

La estructura condicional `\ifx` compara dos tokens hasta su expansión de primer nivel. Si los dos tokens tienen la misma expansión de primer nivel, entonces se ejecuta el bloque verdadero, de lo contrario se ejecuta el bloque falso. Sintaxis de `\ifx`:

```
\ifx⟨token⟩⟨token⟩
  ⟨true code⟩
\else
  ⟨false code⟩
\fi
```

Esto se muestra si los dos tokens tienen la misma expansión.

Ejemplo 13.28

```
\def\empty{\empty}
\ifx\empty\empty %
  Esto se muestra si los dos tokens tienen la misma
   ↪ expansión.
\else
  Esto se muestra si los dos tokens se expanden a algo
   ↪ diferente.
\fi
```

Esto se muestra si los dos tokens tienen la misma expansión.

En este ejemplo, se define un token `\empty` que es un reemplazo para `\empty`, y se compara si estos dos tokens son iguales en su expansión de primer nivel. Aunque la definición del token es absurda, `\ifx` solo realiza la expansión de primer nivel y la comparación, por lo que la afirmación es verdadera.

Sin embargo, en el siguiente ejemplo:

Ejemplo 13.29

```
Esto se muestra si los dos tokens se expanden a algo diferente.
\def\empty{\relax}
\ifx\empty\relax %
  Esto se muestra si los dos tokens tienen la misma expansión.
\else
  Esto se muestra si los dos tokens se expanden a algo diferente.
\fi
```

Esto se muestra si los dos tokens se expanden a algo diferente. Esto se muestra si los dos tokens se expanden a algo diferente.

Aunque `\empty` se define como un reemplazo para `\relax`, cuando se expande, tiene un significado diferente al de `\relax`. Por lo tanto, estos dos tokens son diferentes, y la afirmación es falsa.

La estructura condicional \if compara dos tokens expandiéndolos hasta que encuentra los dos siguientes tokens inexpandibles. Si estos dos tokens son iguales, entonces se ejecuta el bloque verdadero, de lo contrario se ejecuta el bloque falso. Un ejemplo útil es cuando se sabe que una macro contiene a lo sumo un solo carácter y se quiere probar si contiene uno en particular:

Ejemplo 13.30

```
\def\choice{a}
\if b\choice
  Esto se muestra para la opción 'b'.
\else
  Esto se muestra para todas las demás opciones.
\fi
```

Esto se muestra para todas las demás opciones.

La estructura condicional \iftrue siempre ejecuta el bloque verdadero, mientras que la estructura condicional iffalse siempre ejecuta el bloque falso.

La estructura condicional \ifcase ejecuta un bloque de código dependiendo del valor de un número entero o registro de número entero. Sintaxis de \ifcase:

```
\ifcase⟨expression⟩
  ⟨case 0 code⟩
\or
  ⟨case 1 code⟩
\or
...
\or
  ⟨case n code⟩
\else
  Ningun caso
\fi
```

Otros condicionales:

1. \ifodd: comprueba si un número es impar.
2. \ifhmode: comprueba si el modo actual es horizontal.
3. \ifvmode: comprueba si el modo actual es vertical.
4. \ifmmode: comprueba si el modo actual es matemático.

Variables booleanas

La expresión \newif⟨if-name⟩ se utiliza para declarar una nueva "variable booleana". Por ejemplo, se puede declarar una variable booleana llamada "marginpar" con la instrucción \newif\ifmarginpar. A

partir de ahí, se pueden usar los comandos `\marginpartrue` y `\marginparfalse` para asignar valores booleanos a la variable, y se puede utilizar la instrucción `\ifmarginpar` para comprobar su valor.

Es importante destacar que el nombre de la variable booleana que se quiere crear debe empezar con `\if`. Esto se debe a que, de esta manera, se permite la anidación de pares `\if...\fi`.

En este caso, si la variable booleana "modo vuelo" es verdadera, se ejecutará el código que se encuentra en el primer bloque, mientras que si es falsa, se ejecutará el código del segundo bloque.

Por ejemplo, si se quiere determinar si estamos en la mainmatter de nuestro documento LaTeX, podemos crear un nuevo variable, booleana de nombre `@mainmatter`. Un comando específico como el comando chapter puede evaluar si este comando se está invocando en la parte de mainmatter o en otra parte, ya dependiendo del resultado el comando chapter se puede comportar de distinta forma.

Ejemplo 13.31

```
\makeatletter
\newif\if@mainmatter\@mainmattertrue
\if@mainmatter
  Este código se ejecuta si estamos en la mainmatter
\else
  Este código se ejecuta si no estamos en la mainmatter
\fi
\makeatother
```

Este código se ejecuta si estamos en la mainmatter

Bucles

Como has visto, en TEX tenemos un control muy específico sobre la expansión de tokens. Esto hace posible construir incluso bucles a través de medios de recursión. En esencia, un bucle consta de las siguientes partes:

1. Contador o, más generalmente, lista de elementos
2. Incrementador, o más generalmente, selector del próximo elemento
3. Umbral o, más generalmente, marcador de fin de lista
4. Una comprobación del umbral o marcador de fin, respectivamente

Mostraremos dos casos de bucles, el bucle de conteo y el bucle sobre una lista de elementos.

Los bucles de conteo requieren de un contador (`\count0`), un incrementador (`\advance`), un umbral (en este caso 3) y una comprobación (`\ifnum\count=10`) para saber si se ha alcanzado el límite. El siguiente código muestra cómo implementar un bucle de conteo:

Ejemplo 13.32

```
\long\def\countingloop#1 in #2:#3#4{%
#1=#2 %
\loopcounter{#1}{#3}{#4}%
}
\long\def\loopcounter#1#2#3{%
  #3%
  \ifnum#1=#2 %
  \else%
    \advance#1 by1 %
    \loopcounter{#1}{#2}{#3}%
  \fi%
}
\countingloop{\count0} in 0:{3}{%
El valor actual es: ‘\the\count0’\par
}
```

El valor actual es: ‘0’
El valor actual es: ‘1’
El valor actual es: ‘2’
El valor actual es: ‘3’

Es importante tener en cuenta ciertos detalles:

1. Se han utilizado muchos signos % en el código para indicarle a TEX que debe dejar de buscar más dígitos, ya que continuará escaneando token por token hasta estar seguro de que el número ha finalizado.

2. El umbral (3) se ha incluido entre llaves para indicarle a TEX que se trata de un único argumento, ya que, de lo contrario, TEX reconocerá solo el primer dígito como argumento.

3. Debido a que los bucles se realizan por recursión, se deben tener en cuenta las implicaciones de tener muchos `\fis` (if al revés). Para evitar esto, se ha introducido una nueva macro llamada `\next` que determina si el bucle debe continuar o finalizar.

4. No es buena práctica utilizar uno de los contadores del sistema (`\count0`), ya que este podría ser utilizado en otro lugar y provocar errores en el bucle. Por esta razón, se recomienda crear un nuevo contador.

Ejemplo 13.33

```
\long\def\countingloop#1 in #2:#3#4{%
#1=#2 %
\loopcounter{#1}{#3}{#4}%
}
\long\def\loopcounter#1#2#3{%
```

```
  #3%
  \ifnum#1=#2 %
    \let\next=\relax%
  \else
    \advance#1 by1 %
    \def\next{\loopcounter{#1}{#2}{#3}}%
  \fi
  \next
}
\newcount\ourcounter
\countingloop{\ourcounter} in 0:{3}{%
The current value is ‘\the\ourcounter’\par
}
```

The current value is ‘0’
The current value is ‘1’
The current value is ‘2’
The current value is ‘3’

En resumen, el bucle funciona de la siguiente manera:

1. Se expande el comando `\countingloop` y se asigna el valor inicial al contador.
2. Se establece el valor inicial del contador en 0.
3. Se expande el comando que imprime el valor actual del contador y se crea un if statement.
4. Se imprime el valor actual del contador.
5. Se compara el valor actual del contador con el umbral.
6. Si el valor es falso, se incrementa el contador en 1 y se establece `\next` para que el bucle continúe.
7. Si el valor es verdadero, se establece `\next` en `\relax` para que el bucle finalice.
8. Se vuelve a evaluar el bucle, se incrementa el contador y se establece `\next` para que el bucle continúe.

Iterar sobre una lista de elementos es muy similar, solo que en lugar de `\ifnum` necesitamos usar `\ifx` y necesitamos algún marcador de finalización en lugar del valor umbral. Sin embargo, ¿cómo especificamos la lista en sí? Crearemos una lista separada por comas, por ejemplo `{a,b,c,d}` y llamaremos al marcador de finalización `\listingloopENDMARKER`. Explicación del código:

1. Hemos definido `\listingloopENDMARKER` para que se reemplace a sí mismo. Esto es posible porque `\ifx` solo comparará la expansión de primer nivel.
2. También parece que falta un espacio en `...#1in#2...` Sin embargo, los tokens siempre terminan con un espacio en blanco adicional, por lo que no se necesita ninguno aquí y más de un espacio en blanco probablemente se eliminará.

3. La definición `\looppicker#1#2#3,...` tiene tres argumentos, ¡pero la llamada recursiva `\looppicker{#1}{#2}` solo da dos argumentos! ¡Esto es lo que hace posible este tipo de lista! TEX en realidad está escaneando más allá del alcance del token dado para obtener el tercer argumento. En efecto, estamos mordiendo pieza por pieza, elemento de la lista por elemento de la lista de la lista dada. Todo porque hemos declarado una coma adicional, que es el separador de elementos, en la definición de la macro `\looppicker`. La expansión de la macro de bucle siempre recogerá un elemento más de la lista concatenada al final hasta que haya alcanzado el `\ENDMARKER`. Esto se agrega al final de la lista al comienzo del bucle y ahí se detiene.

El macro `\loop` es una macro que se puede utilizar para crear bucles de conteo. Sintaxis de la macro:

```
\loop ParteA \if... ParteB \repeat
```

Donde `ParteA` y `ParteB` son conjuntos de comandos. Primeramente, se procesa la parte A luego se evalua la condición, si la condición es verdadera, entonces se procede a procesar la parte B y luego nuevamente vuelve a procesar la parte A, y si la condición es falsa, inmediatamente se sale del bucle sin procesar la parte B.

Ejemplo 13.34

```
\newcount\minum
\def\numeros#1{
  \ifnum#1< 1%
  \else
  1%
  \minum=1%
  \loop
  \advance\minum by 1%
  \ifnum\minum<#1%
    ,\the\minum
    \repeat
  \fi
}
\numeros{20}
```

1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20

Bucles LaTeX:

1. `\@whilenum{TestNum} \do {Acción}`: Ejecuta la acción mientras la condición sea verdadera.

2. `\@whiledim{TestLong} \do {Acción}`: Ejecuta la acción mientras la condición sea verdadera.

Ejemplo 13.35

```
\newcount\minum
\newcount\cuenta
\makeatletter
\def\pares#1{%
  \minum=2%
  \@whilenum\minum<#1\do{
    \the\minum, \advance\minum by 2%
  }
}
\makeatother
Los números pares menores que 95
son: \pares{95}
```

Los números pares menores que 95 son: 2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46, 48, 50, 52, 54, 56, 58, 60, 62, 64, 66, 68, 70, 72, 74, 76, 78, 80, 82, 84, 86, 88, 90, 92, 94,

13.4 Introducción a tokens de caracteres

Cuando TeX analiza la entrada, asigna a cada carácter leído un código de categoría. La interpretación de la entrada por parte de TeX depende tanto del carácter como de su código de categoría. El programador tiene la capacidad de establecer 16 tokens de caracteres, además de un token especial interno. Los 16 tokens se enumeran desde 0 en adelante. El token de categoría 0 se reserva para caracteres de escape. Los tokens de caracteres son:

1. Inicio de grupo: {
2. Fin de grupo: }
3. Inicio de modo matemático: $
4. Alineación: &
5. Fin de línea
6. Parámetro para macros: #
7. Superíndice matemático: ^
8. Subíndice matemático: _
9. Ignorado completamente
10. Espacio
11. Letras: el alfabeto.
12. 'Otros' caracteres - todo lo demás: ., 1, :, etc.
13. Carácter activo - se interpreta como secuencias de control: ~
14. Inicio de comentario: %
15. Inválido en la entrada: [DEL]

Cuando TeX lee la entrada, cada carácter se asocia con un código de categoría para generar tokens. Por lo tanto, si la entrada es

```
$ 1^{23}_a $
```

TeX lee:

- Un token de cambio matemático y entra en modo matemático
- Un espacio, que se ignora en modo matemático
- Un token "otro", que simplemente se escribe aquí 1
- Un token de superíndice matemático, lo que significa que el siguiente elemento será superíndice
- Un token de grupo de inicio,
- Los tokens "other" y , que no se pueden componer hasta que finalice el grupo 23
- Un token de cierra grupo, que permite a TeX componer el superíndice
- Un token de subíndice matemático, por lo que se mueve el siguiente elemento a una posición de subíndice
- La letra a, que sin significado especial es tipográfica a
- Un espacio, de nuevo ignorado
- Un token de cambio matemático y vuelve al modo horizontal

Los códigos de categoría a menudo son importantes cuando TeX está decidiendo qué es y qué no es una secuencia de control. Con solo el alfabeto como 'letras', algo como

```
\hello@
```

es la secuencia de control `\hello` seguida del token 'otro' `@`. Por otro lado, si hago que `@` sea una letra

```
\catcode`@=11\relax
\hello@
```

entonces TeX buscará una macro llamada `\hello@`. Esto se usa comúnmente en el código LaTeX para aislar las macros privados de las macros de usuario. Así es como se encuentran las macros como `\@for`. La idea de esto es "proteger al usuario de sí mismo".

Podemos usar diferentes tokens de categoria para crear áreas de código 'privadas'. Por ejemplo, LaTeX2e usa `@` como una letra extra y mientras que LaTeX3 utiliza `:` y `_`. Eso efectivamente aísla el código interno LaTeX3 y LaTeX2e cuando se usan juntos (como ocurre en la actualidad).

13.5 Estructura de clase

La estructura de clase para nuestro libro será:

```
\NeedsTeXFormat{LaTeX2e}
\ProvidesClass{artbook}[2023/04/21 v1.0 LaTeX class for book]
\LoadClass{book}
```

```
%-----------------------------------%
%------------- Módulos -------------%
%-----------------------------------%
%------------------------------------------%
%------------- Código inicial -------------%
%------------------------------------------%
%cargado de paquetes
%comandos iniciales
%booleanos
%cajas
%dimensiones
%---------------------------------------------------%
%------------- Declaración de opciones -------------%
%---------------------------------------------------%
%------- Tamaño de página -------%
%------- Tamaño de letra -------%
%------- Impresión de una cara o de 2 caras -------%
%------- Opciones de borrador -------%
%------- Opciones de portada o titlepage -------%
%------- Openright option -------%
%------- Impresión de 2 columnas -------%
%------- Numeración de ecuaciones a la izquierda -------%
%------- Flush left displays -------%
%------- Open bibliography -------%
%-------------------------------------------------%
%------------- Ejecución de opciones -------------%
%-------------------------------------------------%
%---------------------------------------------------------%
%------------- Paquetes utilizado o cargados -------------%
%---------------------------------------------------------%
%------------------------------------------------%
%------------- Diseno del documento -------------%
%------------------------------------------------%
%------- Fuentes -------%
%------- Párrafos -------%
%------- Page layout -------%
%---- Separación vertical
%---- Dimensión del texto
%---- Márgenes
%---- Notas en pie de página
%---- Float placement parameters
%------- Estilo de página -------%
```

```
%---- Convensión de marcado -------%
%---- Definiendo estilo de página -------%
%--------------------------------------------------%
%------------- Marcado de un documento -------------%
%--------------------------------------------------%
%------- El título -------%
%------- Capítulos y secciones -------%
%---- Building blocks
%---- Comandos para marcado
%---- Definir contadores
%---- Frontmatter, mainmatter y backmatter
%---- Partes
%---- Capítulos
%chapter
%---- Encabezados de nivel inferior
%section
%fragment
%subsection
%subsubsection
%------- Listas -------%
%---- General list parameters
%---- Enumerate
%---- Itemize
%---- Description
%------- Definición de nuevos entornos -------%
%---- Resumen
%---- Versículo
%---- Quotation
%---- Quote
%---- Teorema
%---- Titlepage
%---- Appendix
%------- Definición de nuevos comandos -------%
%------- Configuración de parámetros para entornos existentes -------%
%---- Matrix and tabular
%---- Tabbing
%---- Minipage
%---- Framed boxes
%---- Equation and eqnarray
%------- Objetos flotantes -------%
%---- Figure
%---- Table
```

```
%---- Subtítulos
%------- Cambio de fuente -------%
%-----------------------------------------------%
%------------- Referencias cruzadas -------------%
%-----------------------------------------------%
%------- Tabla de contenido y otros -------%
%---- Table of contents
%chapter
%section
%fragment
%subsection
%subsubsection
%---- List of figures
%---- List of tables
%------- Biografía -------%
%------- Index -------%
%------- Notas al pie -------%
%-----------------------------------------%
%------------- Inicialización -------------%
%-----------------------------------------%
%------- Palabras -------%
%------- Fecha -------%
%------- Modo dos columnas -------%
%------- Estilo de página -------%
%------- Impresión de 2 caras y una sola cara -------%
%------- Comando-------%
```

13.6 Archivo main y artbook

Como ya dijimos anteriormente, el archivo main no es más que un documento LaTeX. En este documento se contiene el siguiente código:

```
\documentclass{artbook}
\begin{document}
\chapterfile{PRUEBA}
\end{document}
```

En el archivo "TEORIA-PRUEBA.tex" se contiene:

```
\chapter{PRUEBA}
\section{Sección}
\marginpar{%
  Notas en el margen
```

```
  Notas en el margen
  Notas en el margen
  Notas en el margen
}
\lipsum[1]
\subsection{Subsección}
\lipsum[1]
\subsubsection{Subsubsección}
\lipsum[1]
\section{Sección}
\lipsum[1]
```

Ahora comenzaremos a desarrollar el código de la clase artbook. Primero definiremos el nombre de la clase, la fecha de creación y la versión de la clase. El código es el siguiente:

```
\NeedsTeXFormat{LaTeX2e}
\ProvidesClass{artbook}[2023/04/21 v1.0 LaTeX class for book]
\LoadClass[10pt]{book}
```

Utilizamos el comando `\NeedsTeXFormat` para indicar que la clase artbook es compatible con LaTeX2e. El comando `\ProvidesClass` define el nombre de la clase, la fecha de creación y la versión de la clase. El comando `\LoadClass` carga la clase `book`.

13.7 Módulos de clase

```
%-----------------------------------%
%------------- Módulos -------------%
%-----------------------------------%
```

Los módulos son archivos que contienen código LaTeXque se pueden importar en un documento LaTeX. Los módulos son útiles para organizar el código LaTeX, ya que se pueden importar en un documento LaTeX. Los módulos se pueden importar en un documento LaTeXutilizando el comando `\input`.

Un módulo de una clase puede contener solamente los estilos de página y otro módulo puede tener el estilo de secciones. Para esta plantilla no crearemos ningún módulo.

13.8 Código inicial

```
%------------------------------------------%
%------------- Código inicial -------------%
%------------------------------------------%
```

Cargado de paquetes

Los paquetes son esenciales en LaTeX, ya que nos permiten personalizar nuestro documento de acuerdo a nuestras necesidades y agregar funcionalidades que de otra manera serían difíciles de lograr. Ahora aprenderemos cómo utilizar algunos de los paquetes más útiles para la creación de documentos avanzados, como el paquete titlesec y entre otros.

Los paquetes que estaremos utilizando serán los siguientes:

```
%cargado de paquetes
\RequirePackage{xparse}
\RequirePackage[x11names,table,svgnames,usenames,dvipsnames]{xcolor}
\RequirePackage{amsmath,amsfonts,amssymb,amsthm,latexsym,cancel}
\RequirePackage{amscdx,delarray,empheq}
\RequirePackage{pdfpages}
\RequirePackage[pagestyles,explicit]{titlesec}
\RequirePackage{titletoc}
\RequirePackage{colors}%colores definidos por usuario
\RequirePackage[spanish,es-tabla,es-noshorthands]{babel}
\RequirePackage{tikzpagenodes}%nodos de una pagina (current page)
\RequirePackage{tikz}
\usetikzlibrary{backgrounds}
\usetikzlibrary{shapes.misc}
\usetikzlibrary{fit}
\usetikzlibrary{shadows.blur}
\RequirePackage{multicol}
\RequirePackage{lipsum}
\RequirePackage[
  breaklinks=true,
  bookmarks=true,
  colorlinks=true,
  linkcolor=black,
  citecolor=black,
  filecolor=r7,
  urlcolor=b7,
  anchorcolor=b7,
  pdfstartview=FitV,
  frenchlinks=true
]{hyperref}
\hypersetup{
  pageanchor=true,
  filebordercolor=black,
  linkbordercolor=black,
```

```
  menubordercolor=black,
  pdfstartpage=1,
  pdfpagetransition=Split,
  pdfwindowui,
}
\RequirePackage{bookmark}
\bookmarksetup{open,numbered}
\RequirePackage{sidenotes}
\RequirePackage{anyfontsize}
\RequirePackage[utf8]{inputenc}
\RequirePackage[T1]{fontenc}
\RequirePackage{newtxmath}
\RequirePackage{tgtermes}
\RequirePackage[onehalfspacing]{setspace}
```

Función de los paquetes utilizados:

1. `xparse`: se utiliza para definir nuevos comandos y ambientes.
2. `xcolor`: proporciona un amplio rango de colores para personalizar documentos.
3. `amsmath,amsfonts,amssymb,amsthm,latexsym,cancel`: son paquetes matemáticos que ofrecen funciones avanzadas para escribir fórmulas matemáticas.
4. `amscdx,delarray,empheq`: son paquetes para mejorar la escritura de fórmulas matemáticas.
5. `pdfpages`: permite incluir páginas de documentos PDF existentes en un documento LaTeX.
6. `titlesec`: proporciona herramientas para personalizar el estilo de los títulos de sección.
7. `titletoc`: permite personalizar el formato del índice.
8. `colors`: define colores personalizados.
9. `babel`: se utiliza para escribir documentos en español, permite la configuración de reglas de idioma.
10. `tikzpagenodes`: proporciona herramientas para dibujar figuras en el fondo de la página.
11. `tikz`: proporciona herramientas para dibujar gráficos y figuras.
12. `multicol`: permite dividir el texto en varias columnas.
13. `lipsum`: se utiliza para generar texto de ejemplo.
14. `hyperref`: se utiliza para añadir hipervínculos a los documentos, tanto dentro del mismo documento como hacia otros documentos.
15. `bookmark`: se utiliza para personalizar los marcadores (bookmarks) en PDF.

16. `sidenotes`: permite añadir notas en el margen.
17. `anyfontsize`: permite utilizar cualquier tamaño de fuente.
18. `inputenc`: permite utilizar caracteres Unicode en el texto de entrada.
19. `fontenc`: permite utilizar caracteres acentuados y especiales.
20. `newtxmath`: proporciona un nuevo juego de fuentes matemáticas.
21. `tgtermes`: proporciona un nuevo juego de fuentes de texto.
22. `setspace`: se utiliza para establecer el espacio entre líneas en el documento.

`colors.sty`

El archivo `colors.sty` contiene los colores definidos por el usuario. El código es el siguiente:

```
\definecolor{pu0}{HTML}{F3E5F5}
\definecolor{pu5}{HTML}{9C27B0}
\definecolor{r1}{HTML}{FFCDD2}
\definecolor{i0}{HTML}{E8EAF6}
\definecolor{i1}{HTML}{C5CAE9}
\definecolor{i4}{HTML}{5C6BC0}
\definecolor{g1}{HTML}{C8E6C9}
\definecolor{g2}{HTML}{A5D6A7}
\definecolor{b1}{HTML}{90CAF9}
\definecolor{b4}{HTML}{2196F3}
\definecolor{t1}{HTML}{B2DFDB}
\definecolor{t0}{HTML}{E0F2F1}
\definecolor{t2}{HTML}{80CBC4}
\definecolor{dp0}{HTML}{EDE7F6}
```

Estas lineas define los diferentes colores que se utilizarán en la plantilla.

Comandos iniciales

```
%comandos iniciales
\long\def\@title{Título}
\long\def\@subtitle{Subtítulo}
\long\def\@author{Author}
\long\def\@date{2023}
```

Estos comandos definen el título, subtítulo, autor y fecha del documento.

Booleanos, cajas y dimensiones

```
%booleanos
\newif\if@restonecol%tmp for one column
\newif\if@resttwocol%tmp for two column
\newif\if@titlepage
\newif\if@openright
\newif\if@mainmatter \@mainmatterfalse
\newif\if@backmatter \@backmatterfalse
\@twosidetrue
\@mparswitchtrue %marginote en en margen exterior
\@titlepagetrue
\@openrightfalse
\@twocolumnfalse
%cajas
\newsavebox\@arttempboxa
%dimensiones
\newdimen\@arttempdima
\newdimen\@arttempdimb
```

En cuanto a las variables booleanas, se definen:

1. `\if@restonecol`: se utiliza como una variable temporal para indicar si se ha terminado una página en modo de una columna.
2. `\if@resttwocol`: se utiliza como una variable temporal para indicar si se ha terminado una página en modo de dos columnas.
3. `\if@titlepage`: se utiliza para indicar si se va a crear una página de título en el documento.
4. `\if@openright`: se utiliza para indicar si los capítulos deben comenzar en una página derecha.
5. `\if@mainmatter`: se utiliza para indicar si el documento está en la sección principal.
6. `\if@backmatter`: se utiliza para indicar si el documento está en la sección de los apéndices, epílogo ó tabla de contenido.
7. `\@twoside`: se utiliza para indicar si el documento se imprimirá a doble cara.
8. `\@mparswitch`: se utiliza para indicar si las notas al margen se colocarán en el margen exterior.
9. `\@titlepage`: se utiliza para indicar si se va a crear una página de título en el documento.
10. `\@openright`: se utiliza para indicar si los capítulos deben comenzar en una página derecha.
11. `\@twocolumn`: se utiliza para indicar si el documento se imprimirá a doble columna.

También se definen cajas y dimensiones temporales para ser utilizadas en el documento.

13.9 Declaración de opciones y ejecución

```
%------------------------------------------------%
%------------- Declaración de opciones -------------%
%------------------------------------------------%
```

Este documento class no cuenta con opciones por defecto, por lo que no es necesario ejecutar ninguna. No obstante, si se desea añadir opciones, se pueden agregar de manera personalizada.

13.10 Diseño de página

```
%-----------------------------------------------%
%------------- Diseno del documento -------------%
%-----------------------------------------------%
```

Fuentes

```
%------- Fuentes -------%
\RequirePackage{microtype}
\RequirePackage[onehalfspacing]{setspace}
```

Párrafos

```
%------- Párrafos -------%
\lineskip1\p@\relax
\normallineskip1\p@\relax
\parskip6pt\relax
\parindent0pt\relax
```

Page layout

```
%------- Page layout -------%
\RequirePackage{geometry}
\geometry{%
  includehead, includefoot,
  includemp,heightrounded,
  paperwidth=20cm,left=1.5cm,right=1cm,
  marginparsep=5mm, marginparwidth=5cm,
  paperheight=26.25cm,top=5.5mm,bottom=9mm,
}
```

Define el tamaño de las páginas.

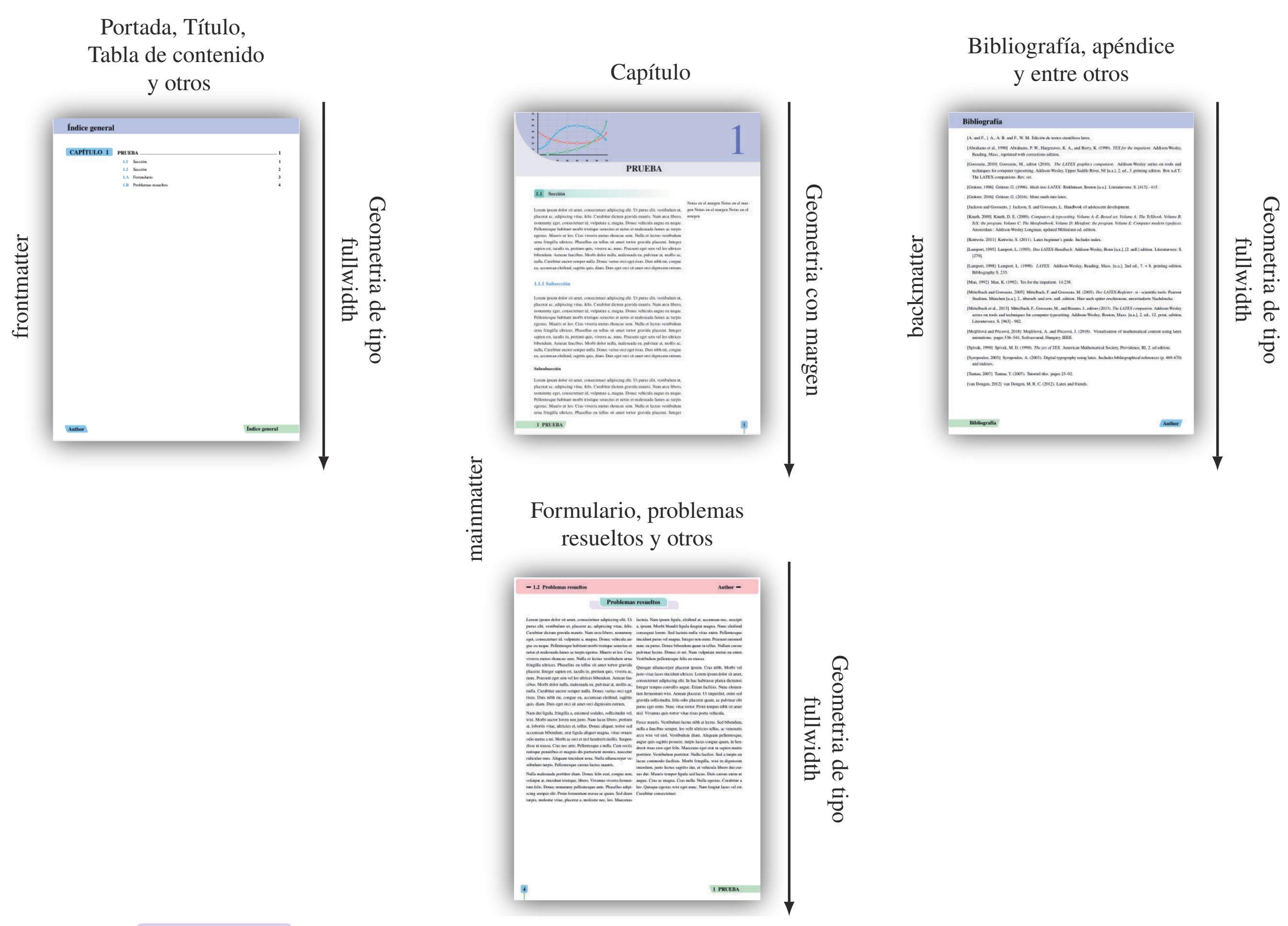


Figura 13.3

Dimensiones geométricas del libro

La plantilla que se desea crear cuenta con dos tipos de geometría de página: uno que incluye márgenes y otro que no los tiene. Aquellas páginas con márgenes se encuentran en el capítulo, mientras que las que no los tienen están en frontmatter, formulario, problemas resueltos, problemas propuestos y backmatter. Por consiguiente, es necesario crear una nueva geometría que permita que el texto ocupe todo el ancho de la página.

Para crear esta nueva geometría se utilizará el comando `\newgeometry`, el cual es proporcionado por el paquete geometry.

```
\newgeometry{
  includehead, includefoot,
  paperwidth=20cm,left=1.5cm,right=1cm,
  paperheight=26.25cm,top=5.5mm,bottom=9mm,
```

```
}
\savegeometry{geometry@full@width}
```

Estilo de página

Se pueden identificar cuatro estilos de página en un documento. El primero, denominado `foot`, se aplica desde la página de derechos de autor hasta el inicio del primer capítulo. Dentro de la sección de frontmatter, habrá una página específica en la que se invocará el comando `\chapter*`, que tendrá su propio estilo, denominado `schapter`. En la sección de mainmatter, se utilizará un estilo de página denominado `all` y en las páginas en las que se invoque el comando `\chapter`, se utilizará un estilo de página llamado `chapter`. En la sección de backmatter se aplicará el mismo procedimiento que en la sección de frontmatter. El comportamiento de los objetos o elementos en los distintos estilos de página varía en función del estilo aplicado.

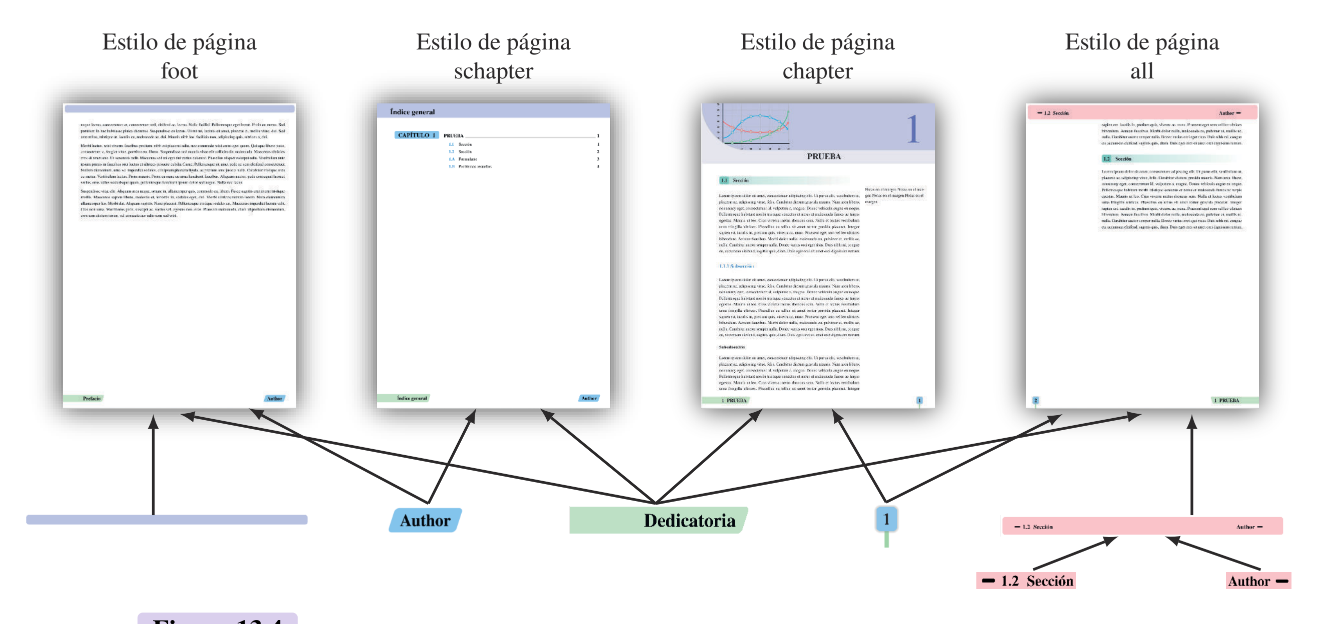


Figura 13.4

Diagrama de estilos de páginas

Definiendo estilo de página

```
%---- Definiendo estilo de página -------%
\newcommand{\@autor@page@style}{
  \if@mainmatter
    \begin{tikzpicture}
      \node[
        inner xsep=1.7mm,
```

```
          outer sep=0pt,
        ](ba)
        {
          \large\bfseries\@author
        };
        \checkoddpage\ifoddpage
          \fill[black,rounded corners=1pt] ([yshift=-0.5mm]ba.west) rectangle ++(-3.8mm,1mm);%
        \else
          \fill[black,rounded corners=1pt] ([yshift=-0.5mm]ba.east) rectangle ++(3.8mm,1mm);%
        \fi
      \end{tikzpicture}
    \else
        \checkoddpage\ifoddpage
        \begin{tikzpicture}[overlay, remember picture]
          \node[
            anchor=east,
            minimum height=6mm,
            outer sep=0pt,
          ](ba)
          {
            \large\bfseries\@author
          };
          \begin{pgfonlayer}{background}
            \fill[b1, rounded corners=3pt] (ba.north west) -- ([xshift=2mm]ba.north east) -- (ba.south east) -- ([xshift=-2mm]ba.south west) -- cycle;
          \end{pgfonlayer}
        \end{tikzpicture}
        \else
          \begin{tikzpicture}[overlay, remember picture]
          \node[
            anchor=west,
            minimum height=6mm,
            outer sep=0pt,
          ](ba)
          {
            \large\bfseries\@author
          };
          \begin{pgfonlayer}{background}
```

```
          \fill[b1, rounded corners=3pt] ([xshift=-2mm]ba.north west) -- (ba.north east) -- ([xshift=2mm]ba.south east) -- (ba.south west) -- cycle;
        \end{pgfonlayer}
        \end{tikzpicture}
      \fi
  \fi
}
\newcommand{\@chapter@page@style}{
  \checkoddpage\ifoddpage
    \tikz[remember picture, overlay]{
      \node[
        anchor=west,
        minimum height=6mm,
        outer sep=0pt,
      ](bc) {
        {\bfseries\large\if@mainmatter\thechapter\;\;\fi\chaptertitle}
      };
      \begin{pgfonlayer}{background}
        \fill[g1] {(bc.north west -| current page.south west)[rounded corners=3pt] -- ([xshift=2mm]bc.north east) -- (bc.south east)} -- (bc.south east -| current page.south west) -- cycle;
      \end{pgfonlayer}
    }
  \else
    \tikz[remember picture, overlay]{
      \node[
        anchor=east,
        minimum height=6mm,
        outer sep=0pt,
      ](bc) {
        {\bfseries\large\if@mainmatter\thechapter\;\;\fi\chaptertitle}
      };
      \begin{pgfonlayer}{background}
        \fill[g1] {(bc.north west -| current page.south east)[rounded corners=3pt] -- ([xshift=-2mm]bc.north west) -- (bc.south west)} -- (bc.south west -| current page.south east) -- cycle;
      \end{pgfonlayer}
    }
  \fi
```

```
}
\newcommand{\@number@page@style}{
  \tikz[remember picture,overlay]{
    \node(bp)[
      fill=b1,
      inner xsep=1.5mm, outer sep=0pt,
      minimum height=6mm,
      anchor=east,
      rounded corners=3pt,
      blur shadow={shadow blur steps=50,shadow blur extra rounding=1.3pt,shadow xshift=0pt, shadow yshift=0pt}
    ]{\textbf{\large\thepage}};
      \begin{pgfonlayer}{background}
        \draw[g2, line width=3pt] (bp.center) -- (bp.center |- current page.south east);
      \end{pgfonlayer}
  }
}
\newcommand{\@section@page@style}{%
  \tikz{%
    \node[%
      line width=0pt,
      inner xsep=1.7mm,
      outer sep=0pt,
    ](sectionPageStyleBox)
    {%
      \large\bfseries\thesection\;\;\sectiontitle
    };
    \checkoddpage\ifoddpage
      \fill[black,rounded corners=1pt] ([yshift=-0.5mm]sectionPageStyleBox.east) rectangle ++(3.8mm,1mm);%
    \else
      \fill[black,rounded corners=1pt] ([yshift=-0.5mm]sectionPageStyleBox.west) rectangle ++(-3.8mm,1mm);%
    \fi
  }
}
\newpagestyle{foot}{%
  \sethead[][][]{}{}{}
  \renewcommand\makeheadrule{%
```

```
      \begin{tikzpicture}[remember picture,overlay]
        \fill[i1, rounded corners=5pt] ([yshift=-1.5mm,xshift=1.5mm]current page.north west) rectangle ([yshift=-8mm,xshift=-1.5mm]current page.north east);
      \end{tikzpicture}%
    }
    \setfoot[\@autor@page@style][][\@chapter@page@style]%  even
            {\@chapter@page@style}{}{\@autor@page@style}%  odd
  }
  \newpagestyle{schapter}{%
    \sethead[][][]{}{}{}
    \setfoot[\@autor@page@style][][\@chapter@page@style]%  even
            {\@chapter@page@style}{}{\@autor@page@style}%  odd
  }
  \newpagestyle{all}{%
    %head
    \sethead[\@section@page@style][][\@autor@page@style]% even
    {\@autor@page@style}{}{\@section@page@style}% odd impar
    \renewcommand\makeheadrule{%
      \begin{tikzpicture}[remember picture,overlay]
        \fill[r1, rounded corners=5pt] ([yshift=-1.5mm,xshift=1.5mm]current page.north west) rectangle ([yshift=-4mm,xshift=-1.5mm]current page.east|-current page header area.south east);
      \end{tikzpicture}%
    }
     \setfoot[\@number@page@style][][\@chapter@page@style]%  even
     {\@chapter@page@style}{}{\@number@page@style}%          odd
  }
  \newpagestyle{chapter}{%
    \sethead[][][]{}{}{}
    \setfoot[\@number@page@style][][\@chapter@page@style]%  even
            {\@chapter@page@style}{}{\@number@page@style}%  odd
  }
```

El código define varios comandos personalizados que se utilizan para diseñar los estilos de página. Estos comandos personalizados emplean el paquete TikZ para crear gráficos y recuadros, y hacen uso de condicionales para determinar si la página es par o impar.

1. `@autor@page@style`:

   - Comienza verificando si se encuentra en la parte principal del documento (mainmatter) o no.

- Si es mainmatter, crea un recuadro que muestra el nombre del autor en negrita y grande. Dependiendo de si la página es par o impar, el recuadro tendrá una pequeña extensión de color negro en la esquina izquierda (páginas impares) o en la esquina derecha (páginas pares).
- Si no es mainmatter, crea un recuadro con fondo de color (b1) y esquinas redondeadas que muestra el nombre del autor. Dependiendo de si la página es par o impar, el recuadro se ancla al este o al oeste.

2. `@chapter@page@style`:
   - Verifica si la página es par o impar.
   - Dependiendo de si la página es par o impar, crea un recuadro con fondo de color (g1) y esquinas redondeadas que muestra el número y el título del capítulo en negrita y grande. El recuadro se ancla a la esquina izquierda (páginas impares) o a la esquina derecha (páginas pares).

3. `@number@page@style`:
   - Crea un recuadro con fondo de color (b1), esquinas redondeadas y una sombra difusa que muestra el número de página en negrita y grande.
   - También dibuja una línea vertical de color (g2) que se extiende desde el centro del recuadro hasta la parte inferior de la página.

4. `@section@page@style`:
   - Crea un recuadro que muestra el número y el título de la sección en negrita y grande. Dependiendo de si la página es par o impar, el recuadro tendrá una pequeña extensión de color negro en la esquina derecha (páginas impares) o en la esquina izquierda (páginas pares).

Además, se definen diferentes estilos de página utilizando estos comandos personalizados:

1. `foot`: Define un estilo de página que utiliza `@autor@page@style` y `@chapter@page@style` para diseñar el pie de página.

2. `schapter`: Similar al estilo 'foot', pero pensado para secciones especiales del documento y es utilizado específicamente para una sola página en donde se invoca el comando `\chapter*`.

3. `all`: Combina `@autor@page@style`, `@section@page@style`, `@number@page@style` y `@chapter@page@s` para diseñar la cabecera y el pie de página. También define una caja horizontal de color (r1) que se extiende a lo largo de la parte superior de la cabecera.

4. `chapter`: Utiliza `@number@page@style` y `@chapter@page@style` para diseñar el pie de página.

Estos estilos de página pueden aplicarse a diferentes partes del documento para lograr una apariencia personalizada y coherente en todo el documento.

13.11 Marcado de un documento

```
%-----------------------------------------------------%
%------------- Marcado de un documento -------------%
%-----------------------------------------------------%
```

Título

```
%------- El título -------%
\def\makefrontcover{\includepdf{images/portada.pdf}}
\def\maketitle{%
  \cleardoublepage
  ~\vspace{0.2\paperheight}
  \begin{center}
    \begin{tikzpicture}
      \node{\bfseries\Huge\color{pu5}\MakeUppercase{\@title}};
    \end{tikzpicture}
  \end{center}
  \vfil
  \begin{center}
    {\Large\bfseries\@author\par}
    {\large\bfseries\@date\par}
  \end{center}
  \vfil
  \begin{center}
    \begin{tikzpicture}
      \node[fill=r1, inner sep = 2mm, rounded
       ↪  corners=1mm]{\Large\bfseries\@subtitle};
    \end{tikzpicture}
  \end{center}
}
\def\makecopyright{%
  \newpage
  \begingroup
  \parskip20pt\relax
  \begin{tikzpicture}
    \node[
      anchor=north west,align=left,
      rounded corners=7pt,
      fill=pu0,
      inner sep=25pt,
```

```
      blur shadow={shadow blur steps=50,shadow blur extra rounding=1.3pt,shadow xshift=0pt, shadow yshift=0pt}
    ] at (current page text area.north west){
      Título de la obra: \@title\\[10pt]
      Autor: Heber Mamani Quispe\\[10pt]
      Edición: \@date\\[10pt]
      Tamaño de libro: 20x26cm\\[10pt]
      Páginas: \pageref{LastPage}\\[10pt]
      Idioma: Español
    };
  \end{tikzpicture}

  \lipsum[1-2]
  \endgroup
}
\def\makededication{%
  \newpage
  \chapter*{Dedicatoria}
  \chaptermark{Dedicatoria}
  \lipsum[1-2]
}
\def\makepreface{%
  \newpage
  \chapter*{Prefacio}
  \chaptermark{Prefacio}

  \lipsum[1-2]
}
\def\makeprologue{%
  \newpage
  \chapter*{Prólogo}
  \chaptermark{Prologo}
  \lipsum[1-2]

  \hfill {\bfseries Nombre del redactor de prólogo}
}
```

Este código define varios comandos personalizados para crear diferentes secciones de un libro o documento, como la portada, el título, la página de derechos de autor, la dedicatoria, el prefacio y el prólogo. A continuación, se explica cada uno de los comandos personalizados en detalle:

1. \makefrontcover:

- Incluye un archivo PDF (images/portada.pdf) como portada del documento.

2 `\maketitle`:

- Limpia las páginas dobles y coloca el título del documento en el centro de la página, utilizando un tamaño de fuente enorme, en negrita, y en color 'pu5' (definido previamente). El título se muestra en mayúsculas.
- Muestra el nombre del autor y la fecha en negrita y con un tamaño de fuente grande y mediano, respectivamente, centrados verticalmente en la página.
- Muestra el subtítulo en un recuadro con fondo de color 'r1' (definido previamente), esquinas redondeadas y tamaño de fuente grande en negrita.

3 `\makecopyright`:

- Crea una nueva página y coloca un recuadro con esquinas redondeadas, fondo de color 'pu0' (definido previamente) y una sombra difusa en la esquina superior izquierda de la página.
- Dentro del recuadro, muestra información sobre la obra, incluyendo el título, el autor, la edición, el tamaño del libro, la cantidad de páginas, y el idioma.
- Incluye dos párrafos de texto de relleno (utilizando el comando `\lipsum` del paquete lipsum) al final de la página.

4 `\makededication`:

- Crea una nueva página y coloca el título "Dedicatoria" en el formato de un capítulo sin número. También establece la marca de capítulo a "Dedicatoria".
- Incluye dos párrafos de texto de relleno.

5 `\makepreface`:

- Crea una nueva página y coloca el título "Prefacio" en el formato de un capítulo sin número. También establece la marca de capítulo a "Prefacio".
- Incluye dos párrafos de texto de relleno.

6 `\makeprologue`:

- Crea una nueva página y coloca el título "Prólogo" en el formato de un capítulo sin número. También establece la marca de capítulo a "Prologo".
- Incluye dos párrafos de texto de relleno.
- Al final de la sección, muestra el nombre del redactor de prólogo en negrita y alineado a la derecha.

Estos comandos personalizados pueden utilizarse para generar las secciones correspondientes en el documento de una manera estilizada y coherente.

Ahora nos vamos a extender documento, Mail y Desconectamos los comandos que comienzan con make.

```
\documentclass{artbook}
\begin{document}
\newgeometry{
  includehead, includefoot,
  paperwidth=20cm,left=1.5cm,right=1cm,
  paperheight=26.25cm,top=5.5mm,bottom=9mm,
}
\savegeometry{geometry@full@width}
\frontmatter
\pagestyle{empty}
\loadgeometry{geometry@full@width}
\makefrontcover
\maketitle
\makecopyright
\makededication
\pagestyle{foot}
\makepreface
\makeprologue
\tableofcontents
\mainmatter
\restoregeometry
\pagestyle{all}
\chapter{PRUEBA}
\section{Sección}
\lipsum[1-10]
\loadgeometry{geometry@full@width}
% \fragment{Formulario}
% \fragment{Problemas resueltos}
% \fragment{Problemas propuestos}
\backmatter
\loadgeometry{geometry@full@width}
\pagestyle{foot}
\nocite{*}
\bibliographystyle{apalike}
\bibliography{bibliografia}
\end{document}
```

Capítulos y secciones

Capítulo

```
%------- Capítulos y secciones -------%
%---- Building blocks
%---- Comandos para marcado
%---- Definir contadores
%---- Frontmatter, mainmatter y backmatter
%---- Partes
%---- Capítulos
%chapter
\let\@chapterimage\relax
\newcommand\chapterimage[1]{%
  \if\relax\detokenize{#1}\relax%
  \else%
    \gdef\@chapterimage{%
      \pgfimage[width=10cm,height=4cm]{#1}
    }%
  \fi%
}
\titleformat{\chapter}[display]{}{}{0cm}{%
  \@openrighttrue
  \begin{tikzpicture}[remember picture,overlay]
    \fill[i0] (current page.north west) rectangle ([yshift=-5.2cm]current page.north east);
    \node[%
      fill=i1,
      text width=2\paperwidth,rounded corners=4cm,
      minimum height=8cm,anchor=center,inner sep=0pt,
      blur shadow={shadow blur steps=100,shadow blur extra rounding=5pt,shadow xshift=0pt, shadow yshift=0pt},
      path picture={
        \node[anchor=south west, minimum height=4cm, minimum width=10cm] at (chaptertop.south west){\@chapterimage};
      }
    ]
    at (current page.north east) (chaptertop){};
    \node[anchor=north east,inner sep=0pt,outer sep=0pt] (chapternum) at ([yshift=-1cm,xshift=-1cm]current page.north east) {\fontsize{100}{0}\color{i4}\selectfont\thechapter};
    \node[anchor=north,inner sep=3mm] at ([xshift=5mm, yshift=-4cm]current page.north) {\bfseries\huge#1};
  \end{tikzpicture}
```

```
}[\thispagestyle{chapter}]
\titlespacing{\chapter}{0pt}{0cm}{2.5cm}
%
\titleformat{name = \chapter, numberless}[display]{}{}{0cm}{%
  \begin{tikzpicture}[remember picture, overlay]
    \fill[i1] (current page.north west) rectangle ([yshift=-3mm]current
     ↪  page.east|-current page header area.south east);
     \node[anchor=west] at (current page header area.west)
      ↪  {\fontsize{18}{0}\bfseries\selectfont#1};
  \end{tikzpicture}
}[\thispagestyle{schapter}]
\titlespacing{name=\chapter, numberless}{0pt}{0cm}{-2cm}
```

Este código define la apariencia de los capítulos en el documento, utilizando el paquete titlesec para personalizar la apariencia del título y el paquete tikz para añadir imágenes y figuras.

1. `\let@chapterimage\relax`: Inicializa la macro @chapterimage como relajada (vacía).
2. `\newcommand\chapterimage[1]`: Define un nuevo comando `\chapterimage` con un argumento, que permite establecer una imagen de capítulo específica.
3. `\titleformat{\chapter}[display]`: Define el formato del título de los capítulos numerados.
4. `@openrighttrue`: Abre el capítulo en una página impar (a la derecha).
5. `tikzpicture`: Crea un diseño personalizado para el título del capítulo.
   - Rellena el área superior de la página con el color ’i0’.
   - Crea un nodo con esquinas redondeadas y un fondo de color ’i1’, y coloca la imagen del capítulo en la parte inferior izquierda del nodo. La imagen se ajusta automáticamente al tamaño especificado.
   - Coloca el número del capítulo en la esquina superior derecha de la página, con un tamaño de fuente de 100 y en color ’i4’.
   - Coloca el título del capítulo en negrita y con un tamaño de fuente enorme, a 5 mm de la izquierda del nodo creado anteriormente.
6. `[\thispagestyle{chapter}]`: Aplica el estilo de página ’chapter’ a la página actual.
7. `\titlespacing{\chapter}{0pt}{0cm}{2.5cm}`: Ajusta el espacio alrededor del título del capítulo.
8. `\titleformat{name = \chapter, numberless}[display]`: Define el formato del título para capítulos con estrella.
9. `tikzpicture`: Crea un diseño personalizado para el título del capítulo con estrella.

- Rellena el área superior de la página con el color ’i1’.
- Coloca el título del capítulo en la esquina superior izquierda de la página, en negrita y con un tamaño de fuente de 18.

10 [`\thispagestyle{schapter}`]: Aplica el estilo de página ’schapter’ a la página actual.

11 `\titlespacing{name=\chapter, numberless}{0pt}{0cm}{-2cm}`:Ajusta el espacio alrededor del título del capítulo sin número.

Encabezado de nivel inferior

section

```
%---- Encabezados de nivel inferior
%section
\def\section@bar{
  \tikz[baseline,trim right=0cm,trim left=0cm,remember picture,overlay] {
    \path [left color=t1!60!t0,right color=white] (1pt,-1ex) rectangle (\textwidth,2.5ex);
  }
}
\titleformat{\section}[hang]{\large\bfseries}{\section@bar}{0cm}{
  \begin{tikzpicture}[baseline,remember picture,overlay]
    \node [fill=t2,anchor= base west,rounded corners,minimum height=3.5ex] (number) at (-9pt,0pt) {\textbf{\large\thesection}};
    \node [anchor=base west] at ([xshift=2mm]number.base east) {\large #1};
  \end{tikzpicture}
}
```

1 Se define una barra degradada de color para la sección.

2 Se establece el formato del título de las secciones.

3 Se crea un diseño personalizado para el título de la sección utilizando un entorno gráfico.

fragmento

```
\titleclass{\fragment}[1]{straight}
\newcounter{fragment}[chapter]
\renewcommand{\thefragment}{\thechapter.\Alph{fragment}}
\titleformat{\fragment}[hang]{\bfseries}{}{0cm}{%
  \hfill
  \begin{tikzpicture}
        \node[
```

```
          chamfered rectangle,
          chamfered rectangle angle=45,
          chamfered rectangle sep=1mm,
          chamfered rectangle corners={north east, south west},
          fill=t1, draw=white, line width=2pt,
          inner xsep=5mm, outer sep=0pt,
          anchor=center,
          name=caja
        ]{\Large #1 };
        \coordinate (a) at ([xshift=-8mm, yshift=-2mm]caja.south west);
        \coordinate (b) at (caja.center);
        \coordinate (c) at ([xshift=8mm]caja.south east);
        \scoped[on background layer]
        \node[
          chamfered rectangle,
          chamfered rectangle angle=45,
          chamfered rectangle sep=1mm,
          chamfered rectangle corners={north east, south west},
          fill=dp0,
          inner sep=0pt, outer sep=0pt,
          anchor=north,
          fit=(a) (b) (c)
        ]{};
  \end{tikzpicture}%
  \hfill
  \sectionmark{#1}%
}
\providecommand*{\toclevel@fragment}{1}
\titlespacing{\fragment}{0cm}{0cm}{0cm}[0pt]
```

1. Se define una nueva clase de título llamada "fragment".
2. Se crea un nuevo contador para los fragmentos, que se reinicia con cada capítulo.
3. Se cambia el formato de numeración del fragmento a "Capítulo.Alfabético".
4. Se define el formato del título de los fragmentos.
5. Se crea un diseño personalizado para el título del fragmento utilizando un entorno gráfico.
6. Se establece el nivel de profundidad de los fragmentos en la tabla de contenidos.
7. Se ajusta el espacio alrededor del título del fragmento.

```
%subsection
\titleformat{\subsection}[hang]
↪ {\large\bfseries\color{b4}}{}{0pt}{\thesubsection\;#1}
\titlespacing{\subsection}{0pt}{0.5cm}{0.4cm}
```

subsubsection

```
%subsubsection
\titleformat{\subsubsection}[hang]{\normalsize\bfseries}{}{0cm}{#1}
\titlespacing{\subsubsection}{0pt}{0.3cm}{0.2cm}
```

Definición de nuevos comandos

```
\newcommand{\chapterfile}[2][2]{%
  \@openrighttrue
  \chapterimage{imagesCapitulo/#2}
  \@section@theory{#2}
  \@section@forms{#2}
  \@section@problems@solved[#1]{#2}
}
\newcommand{\@section@theory}[1]{%
  \widenhead*{0pt}{\marginparsep+\marginparwidth}
  \def\rootimage{#1/IMAGES-TEORIA-#1}
  \input{#1/TEORIA-#1}
}
\newcommand{\@section@forms}[1]{
  \loadgeometry{geometry@full@width}
  \widenhead*{0pt}{0pt}
  \fragment{Formulario}
  \begin{multicols}{2}
    \input{#1/FORMULARIO-#1}
  \end{multicols}
}
\newcommand{\@section@problems@solved}[2][2]{%
  \loadgeometry{geometry@full@width}
  \widenhead*{0pt}{0pt}
  \fragment{Problemas resueltos}
  \begin{multicols}{2}
    \input{#2/PR-#2}
  \end{multicols}
}
```

Explicación de comando `\chapterfile`:

1 Se define un nuevo comando llamado `\chapterfile` que acepta dos argumentos, uno opcional y uno obligatorio. El argumento opcional tiene un valor predeterminado de 2.

2 Se establece que el comienzo de un capítulo siempre esté en una página impar.

3 Se llama al comando `\chapterimage` para establecer una imagen de capítulo utilizando el argumento obligatorio que se pasa al comando `\chapterfile`.

4 Se llama al comando `\@section@theory` y se pasa el argumento obligatorio.

5 Se llama al comando `\@section@forms` y se pasa el argumento obligatorio.

6 Se llama al comando `\@section@problems@solved` y se pasan el argumento opcional y el argumento obligatorio.

Exaplicación de comandos `\@section@theory`, `\@section@forms` y `\@section@problems@solved`:

1 `\@section@theory`:

- Se amplía el encabezado de la página para incluir el margen lateral.
- Se define la ruta de la imagen de la teoría.
- Se incluye el archivo de teoría correspondiente al capítulo actual.

2 `\@section@forms`:

- Se carga una geometría de página completa.
- Se amplía el encabezado de la página a su tamaño original (sin margen lateral).
- Se crea un fragmento con el título "Formulario".
- Se comienza un entorno de dos columnas.
- Se incluye el archivo de formulario correspondiente al capítulo actual.
- Se finaliza el entorno de dos columnas.

3 `\@section@problems@solved`:

- Se carga una geometría de página completa.
- Se amplía el encabezado de la página a su tamaño original (sin margen lateral).
- Se crea un fragmento con el título "Problemas resueltos".
- Se comienza un entorno de dos columnas.
- Se incluye el archivo de problemas resueltos correspondiente al capítulo actual.
- Se finaliza el entorno de dos columnas.

13.12 Referencias cruzadas

Tabla de contenidos

```
%-----------------------------------------------%
%------------- Referencias cruzadas -------------%
%-----------------------------------------------%
%------- Tabla de contenido y otros -------%
%---- Table of contents
\setcounter{tocdepth}{1}%imprime solo hasta secciones (nivel 1)
\contentsmargin{0cm}%
%chapter
\titlecontents{chapter}[10pc]{%
  \addvspace{20pt}
  \begin{tikzpicture}[remember picture, overlay]
    \path[fill=b1, rounded corners=3pt] (-4.2,-.1) rectangle (-0.4,.55);
    \pgftext[left,x=-3.9cm,y=2.5mm]{\Large\bfseries CAPÍTULO\; \thecontentslabel};
  \end{tikzpicture}\large\bfseries
}{}{}{\;\titlerule\;\large\bfseries \thecontentspage}
%section
\titlecontents{section}[13pc]{\addvspace{-3pt}}{\contentslabel[\textbf{\color{b4}\thecontentslabel}]{2pc}}{}{~\hfill\textbf{\thecontentspage}}[]
%fragment
\titlecontents{fragment}[13pc]{\addvspace{-3pt}}{\contentslabel[\textbf{\color{b4}\thecontentslabel}]{2pc}}{}{~\hfill\textbf{\thecontentspage}}[]
%subsection
%subsubsection
```

Este código está configurando la apariencia de la tabla de contenidos en un documento LaTeX. Veamos cada parte:

1. `\setcounter{tocdepth}{1}`: Establece la profundidad de la tabla de contenidos para que solo se muestren las secciones (nivel 1) y no las subsecciones o niveles inferiores.
2. `\contentsmargin{0cm}`: Establece el margen de la tabla de contenidos en 0 cm.
3. Configuración del formato de las entradas de capítulo en la tabla de contenidos:
   (a) `\titlecontents{chapter}[10pc]`: Comienza la configuración de las entradas de capítulo con un espacio horizontal de 10 picas antes del contenido.
   (b) `\addvspace{20pt}`: Añade un espacio vertical de 20 puntos entre las entradas de capítulo.

(c) El código dentro del entorno tikzpicture dibuja un rectángulo azul con esquinas redondeadas y coloca el texto "CAPÍTULO" y el número del capítulo dentro del rectángulo.

(d) `\large\bfseries`: Establece el tamaño y estilo de fuente para el título del capítulo (grande y en negrita).

(e) `{\;\titlerule\;\large\bfseries \thecontentspage}`: Añade una línea horizontal y el número de página después del título del capítulo.

4. Configuración del formato de las entradas de sección en la tabla de contenidos:

(a) `\titlecontents{section}[13pc]{\addvspace{-3pt}}`: Comienza la configuración de las entradas de sección con un espacio horizontal de 13 picas antes del contenido y un espacio vertical de -3 puntos.

(b) `\contentslabel[\textbf{\color{b4}\thecontentslabel}]{2pc}`: Coloca el número de la sección en negrita y con un color específico (b4).

(c) `~\hfill \textbf{\thecontentspage}`: Añade el número de página en negrita después del título de la sección, alineado a la derecha.

5. Configuración del formato de las entradas de fragmento en la tabla de contenidos es similar a las entradas de sección.

Estas configuraciones personalizan la apariencia de la tabla de contenidos, dándole un estilo único a las entradas de capítulo, sección y fragmento.

13.13 Inicialización

Comandos

```
%------- Comando-------%
\AtBeginDocument{%
  \newgeometry{
    includehead, includefoot,
    paperwidth=20cm,left=1.5cm,right=1cm,
    paperheight=26.25cm,top=5.5mm,bottom=9mm,
  }
  \savegeometry{geometry@full@width}
  \frontmatter
  \loadgeometry{geometry@full@width}
  \pagestyle{empty}
  \pagenumbering{Alph}
  \makefrontcover
  \maketitle
  \makecopyright
  \clearpage
```

```
    \pagestyle{foot}
    \makededication
    \makepreface
    \makeprologue
    \tableofcontents
    \mainmatter
    \restoregeometry
    \pagestyle{all}
  }
  \AtEndDocument{%
    \backmatter
    \loadgeometry{geometry@full@width}
    \pagestyle{foot}
    \nocite{*}
    \bibliographystyle{apalike}
    \bibliography{bibliografia}
  }
```

Este código define cómo se verá el documento al principio y al final del mismo. Veamos cada parte:

1. `\AtBeginDocument`: Esta parte del código se ejecuta al inicio del documento.
   - `\newgeometry{}`: Define una nueva geometría de página que incluye el encabezado y el pie de página, con dimensiones específicas de ancho, alto, márgenes izquierdo, derecho, superior e inferior.
   - `\savegeometry{geometry@full@width}`: Guarda la geometría de página actual con el nombre "geometry@full@width".
   - `\frontmatter`: Comienza la sección preliminar del documento (páginas en números romanos).
   - `\loadgeometry{geometry@full@width}`: Carga la geometría de página guardada anteriormente.
   - `\pagestyle{empty}`: Establece el estilo de página sin encabezado ni pie de página.
   - `\pagenumbering{Alph}`: Configura la numeración de las páginas con letras mayúsculas del alfabeto.
   - `\makefrontcover`, `\maketitle`, `\makecopyright`: Crea la portada, el título y la página de derechos de autor.
   - `\clearpage`: Comienza una nueva página.
   - `\pagestyle{foot}`: Establece el estilo de página con solo pie de página.
   - `\makededication`, `\makepreface`, `\makeprologue`: Crea las páginas de dedicatoria, prefacio y prólogo.

- `\tableofcontents`: Genera la tabla de contenidos.
- `\mainmatter`: Comienza la sección principal del documento (páginas en números arábigos).
- `\restoregeometry`: Restaura la geometría de página original.
- `\pagestyle{all}`: Establece el estilo de página con encabezado y pie de página.

2 `\AtEndDocument`: Esta parte del código se ejecuta al final del documento.

- `\backmatter`: Comienza la sección de apéndices y bibliografía.
- `\loadgeometry{geometry@full@width}`: Carga la geometría de página guardada anteriormente.
- `\pagestyle{foot}`: Establece el estilo de página con solo pie de página.
- `\nocite{*}`: Añade todas las referencias del archivo de bibliografía, aunque no se hayan citado en el documento.
- `\bibliographystyle{apalike}`: Establece el estilo de la bibliografía como "apalike" (similar al estilo APA).
- `\bibliography{bibliografia}`: Incluye la bibliografía a partir del archivo "bibliografia.bib".

Este código organiza la estructura del documento, definiendo las secciones preliminares, principales y finales, así como los estilos de página y la numeración.

TEX

14.1 Introducción a macros TeX

En TeX un argumento de macro es el texto que se proporciona al llamar a una macro. Por ejemplo, en la siguiente llamada a macro:

```
\foo{arg1}
```

"arg1" es el argumento de la macro `\foo`.

Por otro lado, un parámetro de macro es una variable que se utiliza en la definición de una macro para representar el argumento que se proporcionará al llamar a la macro. Por ejemplo, en la siguiente definición de macro:

```
\def\foo#1{...}
```

`#1` es un parámetro de la macro `\foo` que representa el argumento que se proporcionará al llamar a la macro.

14.2 El estructura del procesador TeX

TEX procesa una entrada en cuatro niveles (a un nivel también se la conoce como etapa o unidad). Cada uno de los niveles acepta una salida del anterior nivel. La entrada al primer nivel es el archivo de extensión .tex y la salida de la última etapa es el archivo .dvi.

Es necesario aclarar que los niveles de procesamiento se ejecutan uno tras otro, aceptando una salida del nivel anterior. Pero esto en realidad no sucede, todos los niveles están activos simultáneamente y hay una interacción entre ellos.

Los términos utilizados para referirse a los niveles por Donald Knut son: eyes, mouth, stomach y bowels. Traducido en español, estos términos serían: ojos, boca, estómago e intestino.

1. **El procesador de entrada** es una pieza que acepta líneas de entrada del sistema de archivos y las convierte en tokens. Los tokens son objetos internos de tex. Existen los tokens de caracteres que constituyen un texto compuesto y tokens de secuencias de control (comandos para procesar en los siguientes dos niveles). Un documento de entrada es de la siguiente forma:

```
\documentclass{article}
\begin{document}
Hola, mundo!
\end{document}
```

2. **Procesador de expansión**. Los tokens generados en el anterior nivel como macros, condicionales y macros primitivos de TEX están sujetos a ser expandidos. En este proceso se expanden los tokens remplazándose por otros tokens, siempre y cuando existan tokens a expandir.

```
\def\saludo{Hola, mundo!}
\saludo
```

3. **Procesador de ejecución**. En este nivel se procesan las secuencias de control que no son expandibles. Una de las actividades que se realiza en este proceso es realizar los cambios internos de TEX, como las asignaciones, incluyendo las definiciones de macros. Además, en este proceso se realizan las construcciones de las listas horizontales y verticales y matemáticas. Por ejemplo:

```
\newcount\miContador
\miContador=5
\advance\miContador by 3
```

4. **Procesador visual**. En este último proceso se divide las listas horizontales en párrafos y las listas verticales en páginas y las fórmulas se consiguen a partir de las listas matemáticas. La generación o salida del archivo dvi se lleva en este etapa de proceso y además los algoritmos utilizados en esta etapa no son accesibles para el usuario, pero podría ser cambiado por un serie de parámetros.

Procesador de entrada

El procesador de entrada es el que se encarga de convertir los caracteres que se encuentran en el archivo de extensión .tex a tokens. La salida de este procesador es una lista de tokens. Generalmente los tokens entran a una de las dos categorías de tokens: Tokens de caracteres y tokens de secuencias de control. además, se tiene una categoría de tokens de parámetros.

Entrada de caracteres

El texto simple se convierte en tokens de caracteres. TEX ignora algunos caracteres como una fila de espacios que es equivalente a un solo espacio. Además, tex introduce algunos tokens que no corresponden

a ningún carácter de entrada; por ejemplo, el token de espacio al final de una línea y el token `\par` en una línea vacía.

Tokens de caracteres

Existen 16 tokens de caracteres, por lo que no todos los tokens de caracteres se van a formatear. Una de las 16 divisiones de categorías tienen una función determinada Que contiene un carácter; Dos categorías de las 16 se llegan a Formatear; Las categorías restantes contienen a caracteres como `$ $`, `&` y `#`. Un token de caracteres es como un par de números. En donde un número es el código de carácter (en este caso el Código ASCII) Y el otro número es el código de categoría. Es posible cambiar un código de categoría que se asocia con un código de carácter.

Cuando aparece el carácter `\` en el texto, tex lo interpreta como un carácter de escape y su comportamiento a la hora de formación De Tokens es más complicado. Se toma un número de caracteres de la entrada Para construir una secuencia de control logo se agrupa en un solo token.

Al momento de procesar la entrada, tex, reacciona a los códigos de categoría como una máquina que cambia entre tres estado internos: `N`, nueva linea; `M`, medio de línea; S, salto de espacio. Se estudiará detalladamente más adelante estes Estados.

Procesamiento de entrada es como de dos niveles

TeX, antes de generar los tokens de caracteres reemplaza ciertas secuencias de caracteres con otros caracteres; por ejemplo, la secuencia `^^+` es reemplazado por `k`. Entonces, si el procesador recibe una secuencia, como `^^+`, en la entrada, esto se reemplazará por `k` antes de convertir los caracteres en tokens. Por lo que el código

```
\vs^^+ip 5cm
```

tiene el mismo efecto que

```
\vskip 5cm
```

Entonces, podemos decir que esta primera etapa es un procesamiento de entradas que transforma los caracteres a caracteres sin tener en cuenta el código de categoría.

Procesador de expansión

Los toques recibidos en el anterior etapa se expanden en esta etapa uno por uno, hasta que queden solo tokens no expandibles; Por ejemplo, la expansión de macros. La expansión de macros consiste en reemplazar algo por otra cosa, por ejemplo, si un token es el nombre de un macro, entonces se remplazará por el texto de la definición del macro.

Una vez expandido los tokens, se envían los tokens no expandibles al proceso de ejecución.

El comando `\edef` es una variante del comando `\def`, que se utiliza para definir nuevos comandos en TEX. La diferencia entre `\edef` y `\def` es que `\edef` expande los tokens de la definición de la macro antes de almacenarla. Por ejemplo, si se tiene la siguiente definición:

```
\edef\foo{\bar\baz}
```

Entonces, cuando se procesa esta línea, el comando `\bar` se expande (si tiene una definición) antes de almacenar la macro `\foo`.

El comando `\write` se utiliza para escribir el contenido de una macro en el archivo de salida del documento. Por ejemplo, si se tiene la siguiente línea:

```
\write\foo
```

Entonces, el contenido de la macro `\foo` se escribirá en el archivo de salida del documento.

Cuando se procesa una lista de tokens como argumento de un comando, normalmente no se expanden los tokens. Sin embargo, cuando se utiliza el comando `\edef` o `\write`, la lista de tokens se expande de manera similar a si estuviera en el nivel principal del documento, es decir, como si no fuera un argumento de un comando. Esto significa que todos los tokens que tienen una definición se expandirán antes de ser procesados.

El proceso de expansión

Para expandir un token, el procesador de expansión sigue los siguientes pasos:

1. Verifica si el token es expandible o no. Algunos ejemplos de tokens expandibles son las macros, los caracteres activos, los condicionales y algunos comandos primitivos de TEX. Los tokens de caracteres como dólares y llaves, entre otros no son expandibles.
2. Si el token no es expandible, se envía a la lista de tokens que se está construyendo y se pasa al siguiente token.
3. Si el token es expandible, se reemplaza por su expansión. Esto puede ser un simple reemplazo si el token es una macro sin parámetros o un comando primitivo. En otras ocasiones, se necesitarán absorber tokens adicionales de la secuencia para poder formar la expansión completa. Por ejemplo, si el token es una macro con parámetros, se necesitarán tokens adicionales para formar los argumentos correspondientes a estos parámetros.
4. Una vez que se ha expandido el token, se continúa expandiendo la secuencia de tokens desde el primer token de la expansión.

En esta etapa también se evalúan las condicionales y los tokens como `\vskip`, tokens de caracteres como el símbolo de dolar y llaves, entre otros, no son expandibles.

Casos especiales: `expandafter`, `noexpand`, y `the`

`\expandafter`

El comando `\expandafter` es una forma de decirle al procesador de expansión que salte un token y expanda el siguiente. Por ejemplo, si tenemos la secuencia

```
\expandafter<token1><token2>
```

el procesador de expansión reemplazará por

```
<token1><expansion of token2>
```

Si el siguiente token es el comando `\noexpand`, entonces el procesador de expansión no expandirá temporalmente el siguiente token.

El comando `\noexpand` es una forma de decirle al procesador de expansión que ignore temporalmente el siguiente token. Por ejemplo, en la definición de macro

```
\edef\foo{\noexpand\bar}
```

el texto de reemplazo `\noexpand\bar` se expandirá en el tiempo de definición. Cuando la expansión llegue a `\b` considerará como un token no expandible por lo que simplemente pasará a una lista de tokens que se está construyendo.

Llaves en proceso de expansión

Una llave Se pasa como toques de caracteres no expandibles. Por ejemplo, el comando `\romannumeral` se maneja por el procesador de expansión y, cuando se encuentra con una secuencia como esta:

```
\romannumeral1 \number\count2 3 { 4 ...
```

el procesador de expansión expandirá todos los tokens hasta que encuentre las llaves. Si `\count2` tiene el valor de cero, el resultado final será la representación numeral romana 103.

El código,

```
\iftrue {\else }\fi
```

es análogo a

```
\iftrue a\else b\fi
```

El resultado es un token de caracter, indenpendientemente de su categoría de token.

En el proceso de expansión, un par de llaves marca un grupo de tokens Que se pasan como un solo argumento. Una macro como

```
\def\foo#1{...}
```

de un argumento se llama con un solo token,

```
\foo 1 \foo \$
```

o cun grupo de tokens delimitados por llaves

```
\foo {abc}
\foo {ab{bg}c}
```

Cuando se define una macro con parámetros, como en

```
\def\a#1\stop{ ... }
```

los parámetros se usan para representar los argumentos que se proporcionarán al llamar a la macro. Por ejemplo, si llamamos a la macro `\a` de la siguiente manera:

```
\a bc{d\stop}e\stop
```

el argumento de la macro es `bc{d\stop}e`, ya que este es el texto que se encuentra entre la primera aparición de `\a` y la primera aparición de `\stop` que no esté en corchetes.

Es importante tener en cuenta que solo se aceptan con llaves de apertura y cierre en los argumentos de las macros con parámetros. Esto significa que, en el ejemplo anterior, el argumento de la macro `\a` es "`bc{d\stop}e`", ya que las llaves están balanceados o existe uno de apertura y cierre. Sin embargo, si llamamos a la macro de la siguiente manera:

```
\a bc{d\stop e\stop
```

esto no sería válido, ya que las llaves no están balanceados o no existen uno de cierre ya que existe el de apertura.

The visual processor

El procesador de salida de TEX abarca aquellos algoritmos que están fuera de control del usuario como cortado de párrafos, alineado, cortado de páginas, escritura matemática Y generación de los archivos de dvi.

Existen unos parámetros para controlar el funcionamiento de estas partes. Un párrafo que se ha dividido en líneas se agrega a una lista vertical principal como secuencias de cuadros horizontales con un pegamento (glue) intermedio y penalización. El algoritmo de salto de página almacena su resultado en `\box255`. Las fórmulas matemáticas no se pueden dividir en partes.

14.3 Códigos de categoría y estados internos

El sistema TEXlee los caracteres de entrada y luego las asigna con códigos de categoría. El proceso de lectura tiene 3 estados internos y la transición entre estos estados se ven afectados por códigos de categoría.

1. `\endlinechar` es el código de categoría de fin de línea. El valor predeterminado es 13.

2. `\par` es el comando para cerrar un párrafo e ingresar al modo vertical. Se genera con líneas vacías.

3. `\ignorespaces` es un comando que lee y expande hasta que se encuentra algo que no es un token de espacio.

4. `\catcode` se usa para consultar o establecer códigos de categoría.

5. `\ifcat` se usa para probar si dos caracteres tienen el mismo código de categoría.

6. `\char32` es un espacio de control. Inserta la misma cantidad de espacio que un token de espacio cuando `\spacefactor` es 1000.

7. `\obeylines` es una macro en TEX simple para hacer que los finales de línea sean significativos.

8. `\obeyspaces` es una macro en TEX simple para hacer que (la mayoría de) los espacios sean significativos.

Introducción

Un autómata de estado finito (Finite State Automaton, FSA) es un modelo matemático que representa un sistema que puede estar en un número finito de estados en un momento dado. Se utiliza comúnmente en la teoría de la computación y el diseño de sistemas de control.

Un autómata de estado finito simple de 3 estados internos es un FSA con las siguientes características:

1. 3 estados internos: El autómata tiene tres estados diferentes en los que puede estar. Estos estados suelen ser representados por círculos o elipses y etiquetados, por ejemplo, como A, B y C.

2. Transiciones: Las transiciones representan cómo el autómata cambia de un estado a otro en respuesta a una entrada. Estas transiciones son representadas por flechas que conectan los estados y etiquetadas con los símbolos de entrada que provocan el cambio de estado.

3. Simple: En este contexto, "simple" puede referirse a que el autómata no tiene estados finales o de aceptación, lo que significa que su función principal es procesar secuencias de entrada y realizar transiciones entre estados, pero no está diseñado para aceptar o rechazar cadenas de entrada como en un autómata de estado finito determinista (DFA). También podría referirse a la simplicidad en la estructura y reglas de transición.

4. Por lo tanto, un autómata de estado finito simple de 3 estados internos es un modelo con tres estados y un conjunto de transiciones que describen cómo cambia de un estado a otro en respuesta a símbolos de entrada, sin un enfoque específico en la aceptación o rechazo de cadenas de entrada.

Inicialmente el sistema TeX lee todas las entradas desde un archivo o terminal y luego los convierte en tokens. El procesador de entrada es como un autómata de Estado finito simple de tres estados internos.

14.4 Caracteres

Macros que se estudiará en esta sección:

1. `\char`: Este comando permite denotar un carácter de forma explícita para ser compuesto.
2. `\chardef`: Este comando permite definir una secuencia de control como sinónimo de un código de carácter.
3. `\accent`: Este comando permite colocar caracteres acentuados.
4. `\if`: Este comando permite probar la igualdad de los códigos de caracteres.
5. `\ifx`: Este comando permite probar la igualdad tanto de los códigos de caracteres como de los códigos de categoría.
6. `\let`: Este comando permite definir una secuencia de control como sinónimo de un token.
7. `\uccode`: Este comando permite consultar o establecer el código de carácter que es la variante en mayúscula de un código determinado.
8. `\lccode`: Este comando permite consultar o establecer el código de carácter que es la variante en minúscula de un código determinado.
9. `\uppercase`: Este comando permite convertir el argumento ⟨General text⟩ a su forma en mayúscula.
10. `\lowercase`: Este comando permite convertir el argumento ⟨General text⟩ a su forma en minúscula.
11. `\string`: Este comando permite convertir un token en una cadena de uno o más caracteres.
12. `\escapechar`: Este comando permite establecer el número del carácter que se utilizará como carácter de escape cuando se conviertan las secuencias de control en tokens de caracteres. El valor predeterminado en IniTEX es 92

14.5 Códigos de caracteres

El sistema TEX trabaja internamente con enteros (códigos de caracteres) y no así con caracteres.

El sistema TEX utiliza su propio esquema de código de caracteres por lo que cuando se lee un archivo o desde una terminal, esto se convierten en códigos de caracteres de acuerdo con su esquema. La tabla de códigos de caracteres de TEX se basa en ascii de 7 bits para números menores a 128

Para convertir explícitamente entre caracteres (tokens de caracteres) y códigos de caracteres se puede utilizar el carácter de cita izquierda ‘ seguido del caracter o secuencia de control de un solo caracter. Por ejemplo, `\count`a` y `\count`\a` con sinónimos de `\count97`.

`\count` se utiliza para acceder a los registros de conteo en TeX. El uso de `\count` seguido de un número (por ejemplo, `\count97`) se refiere al registro de conteo número 97, no al conteo de apariciones de un carácter en el texto.

La asignación de códigos de categoría podemos realizar de la siguiente manera:

Ejemplo 14.1

```
\pdfpagewidth=11cm \hsize=11cm \hoffset=-1in
\pdfpageheight=1cm \vsize=1cm \voffset=-1in
\count0=`a
\the\count0
\bye
```

97

En este ejemplo, asignamos el código de carácter de "a"(97) al registro de conteo `\count0`. Luego se muestra el valor de `\count0`, que será 97.

Secuencias de control para caracteres

`\char`

El comando `\char` se utiliza para agregar o tipografiar un carácter específico a la lista horizontal actualmente en construcción. Por ejemplo,

Ejemplo 14.2

```
\pdfpagewidth=11cm \hsize=11cm \hoffset=-1in
\pdfpageheight=1cm \vsize=1cm \voffset=-1in
\char98
\bye
```

b

indica que se debe agregar el carácter número 98 de la fuente actual a la lista horizontal. En lugar de la notación decimal, también se pueden utilizar la notación octal o hexadecimal para denotar caracteres. Para la notación octal se utiliza una comilla simple, por ejemplo,

Ejemplo 14.3

```
\pdfpagewidth=11cm \hsize=11cm \hoffset=-1in
\pdfpageheight=1cm \vsize=1cm \voffset=-1in
\char'142
\bye
```

b

y para la notación hexadecimal se utiliza una comilla doble, por ejemplo,

Ejemplo 14.4

```
\pdfpagewidth=11cm \hsize=11cm \hoffset=-1in
\pdfpageheight=1cm \vsize=1cm \voffset=-1in
\char"62
\bye
```

b

Además, el comando `\char` puede usarse junto con el carácter de cita inversa para obtener un carácter específico a través de su código de carácter, por ejemplo,

Ejemplo 14.5

```
\pdfpagewidth=11cm \hsize=11cm \hoffset=-1in
\pdfpageheight=1cm \vsize=1cm \voffset=-1in
\char`b
\bye
```

b

El comando `\let`, por otro lado, se utiliza para crear sinónimos de tokens de carácter. Por ejemplo,

Ejemplo 14.6

```
\pdfpagewidth=11cm \hsize=11cm \hoffset=-1in
\pdfpageheight=1cm \vsize=1cm \voffset=-1in
\let\newcommand=\relax
No hay salida \par
\newcommand
\bye
```

No hay salida

`\let`

hace que `\newcommand` se comporte como el token de carácter `\relax`. La combinación de código de carácter y código de categoría se mantiene sin cambios en el nuevo sinónimo creado con el comando `\let`.

Cuando se utiliza el comando `\def` en TEX, se está definiendo una macro con un nombre dado y un cuerpo que puede estar vacío. Por ejemplo:

```
\def\foo{}
```

Esto define la macro `\foo` con un cuerpo vacío. A pesar de que el cuerpo está vacío, `\foo` todavía "existe" en el sentido de que es una macro definida. Esto significa que se puede verificar si existe utilizando comandos como `\ifdefined` o `\@ifundefined` en LATEX. Sin embargo, la misma funcionalidad se

puede lograr utilizando el comando \let, que también puede asignar un valor vacío a una macro. En lugar de definir una macro, \let asigna un nombre a otra macro o variable ya definida. Por ejemplo:

```
\let\foo\empty
```

Este código tiene el mismo efecto que el código anterior, definiendo la macro \foo con un cuerpo vacío. Pero a diferencia de \def, \let es más eficiente porque apunta a una ubicación de memoria ya utilizada. Además, \let también se puede utilizar para asignar cualquier otra macro o variable existente a un nuevo nombre.

Por otro lado, \let también se puede utilizar para asignar el valor del primitivo \relax a una macro. Por ejemplo:

```
\let\foo\relax
```

Esto hace que \foo sea igual al primitivo \relax, que es una operación que no hace nada. Sin embargo, es importante destacar que \relax no es expandible, lo que significa que no se puede expandir a nada más que a sí mismo. Por lo tanto, si se utiliza \edef para expandir una macro con el valor \relax, se obtiene el nombre de la macro, no su valor. Por ejemplo:

```
\edef\baz{\foo}
```

En este caso, \baz tendría la definición '\foo', en lugar de ser vacío como en el ejemplo anterior. Esto puede ser útil en algunos casos como una forma de evitar temporalmente que una macro haga algo mientras se mantiene su apariencia en otro código. Por ejemplo, se puede utilizar \let para hacer que una macro no haga nada temporalmente durante la depuración:

```
\let\debug\relax
```

En general, la elección entre \def y \let depende del caso de uso específico. Si se desea una macro con un cuerpo vacío que se pueda verificar si está definida o no, se debe utilizar \def. Si se desea asignar el valor de una macro o variable existente a un nuevo nombre, o si se desea asignar el valor de \relax a una macro, se debe utilizar \let.

Composición de caracteres con char

En los ejemplos anteriores ya habíamos compuesto caracteres con el macro \char.

Existe una construcción en TeX que permite que una secuencia de control represente algún código de carácter: el comando \chardef. La sintaxis de este comando es:

```
\chardef⟨control sequence⟩⟨equals⟩⟨number⟩
```

\chardef

El número puede ser una representación explícita o un valor de contador, pero también puede ser un código de carácter obtenido usando el comando de comillas simples. En el formato plain, esta posibilidad se utiliza en definiciones como:

```
\chardef\%=‘\%
```

Que podría haber sido equivalente a:

```
\chardef\%=37
```

Después de este comando, el símbolo de control % utilizado por sí solo es sinónimo de `\char37`, es decir, el comando para componer el carácter 37 (generalmente el símbolo de porcentaje).

Una secuencia de control que se ha definido con un comando `\chardef` también se puede utilizar como ⟨número⟩. Este hecho se utiliza en comandos de asignación como `\newbox`. Los tokens definidos con `\mathchardef` también se pueden utilizar de esta manera.

Definamos una secuencia de control para representar el símbolo de exclamación. Ahora podemos usar la secuencia de control `\exclamacion` para representar el símbolo de exclamación en nuestro documento:

Ejemplo 14.7

```
\pdfpagewidth=11cm \hsize=11cm \hoffset=-1in
\pdfpageheight=1cm \vsize=1cm \voffset=-1in
\chardef\exclamacion=33
¡Hola\exclamacion
\bye
```

Hola!

Tokens de caracteres implícitos: let

`\let`

El comando

```
\let⟨control sequence⟩⟨equals⟩⟨token⟩
```

se utiliza para definir una secuencia de control que es un sinónimo de otra secuencia de control o de un carácter.

Por ejemplo, el siguiente código crea un sinónimo para el carácter de apertura de llave ({) y el carácter de cierre de llave (}):

```
\let\bgroup={
\let\egroup=}
```

Acentos

`\accent`

El comando

```
\accent⟨8-bit number⟩⟨optional assignments⟩⟨character⟩
```

se usa para colocar acentos sobre caracteres en `\TeX`. El comando toma tres argumentos: el número de 8 bits del acento, las asignaciones opcionales para ajustar la fuente, y el carácter que va a llevar el acento. Este último debe ser un carácter de categoría 11 o 12, un comando `\char` con el número de carácter correspondiente, o un token `\chardef`. Si no se utiliza uno de estos tres tipos de caracteres después del comando `\accent`, TEXtomará el acento como un comando `\char` suspendido en el aire. La anchura del carácter con acento es la misma que la del carácter sin acento. TEXasume que el acento en el archivo de la fuente está posicionado adecuadamente para un carácter que tiene la misma altura que la altura de x en la fuente. Si el carácter tiene otra altura, el acento se sube o se baja en consecuencia.

Ejemplo 14.8

```
\pdfpagewidth=11cm \hsize=11cm \hoffset=-1in
\pdfpageheight=1cm \vsize=1cm \voffset=-1in
\accent23a
\bye
```

å

Este comando produce una 'e' con un acento agudo encima. El número 23 corresponde al número de 8 bits del acento agudo.

Ejemplo 14.9

```
\pdfpagewidth=11cm \hsize=11cm \hoffset=-1in
\pdfpageheight=1cm \vsize=1cm \voffset=-1in
\accent96{\it a}
\bye
```

‘*a*

Ejemplo 14.10

```
\pdfpagewidth=11cm \hsize=11cm \hoffset=-1in
\pdfpageheight=1cm \vsize=1cm \voffset=-1in
\accent94\char111
\bye
```

ô

Ejemplo 14.11

```
\pdfpagewidth=11cm \hsize=11cm \hoffset=-1in
\pdfpageheight=1cm \vsize=1cm \voffset=-1in
```

```
\chardef\ntilde=241
\accent7\ntilde
\bye
```

ϒ

Testeo de caracteres

\if

El comando

`\if⟨token1⟩⟨token2⟩`

se utiliza para probar si dos tokens son iguales o no.

\ifx

El comando

`\ifx⟨token1⟩⟨token2⟩`

se utiliza en TEX para comparar dos tokens (que pueden ser cualquier tipo de token, como una letra, una caja, un contador, un comando, etc.) para determinar si son iguales. A diferencia de `\if`, que se utiliza para comparar códigos de caracteres, `\ifx` compara tanto el código de carácter como la categoría del token.

Ejemplo 14.12

```
\pdfpagewidth=11cm \hsize=11cm \hoffset=-1in
\pdfpageheight=1cm \vsize=1cm \voffset=-1in
\def\a{foo}
\def\b{\a}
\ifx\a\b
Verdadero
\else
Falso
\bye
```

Falso

Al comparar `\a` y `\b` con `\ifx`, se obtiene un resultado falso porque sus caracteres con distintos.

Códigos de caracter y codename

Cada código de carácter en TEX tiene asociados varios números de código a las cuales se las conoce como codenames. Estos son números enteros en diferentes rangos que determinan cómo se trata el carácter en varios contextos o cuando aparecen en ciertos lugares.

1 `\catcode<4 bit number> (0-15)`: El código de categoría determina cómo se tratan los caracteres en diferentes contextos en TEX. Hay 16 categorías posibles (0-15). Por ejemplo, las letras son de categoría 11 y los números de categoría 12.

Ejemplo 14.13

```
\pdfpagewidth=11cm \hsize=11cm \hoffset=-1in
\pdfpageheight=1cm \vsize=1cm \voffset=-1in
\catcode`@=11 % Cambia la categoría del carácter '@' a letra
\def\saludo@amigo{¡Hola, amigo!}
\saludo@amigo
\bye
```

Hola, amigo!

2 `\mathcode<15 bit number> (0-"7FFF) o 8000`: El código matemático determina cómo se trata un carácter en modo matemático. Por ejemplo, si un carácter es tratado como una letra, un operador, etc.

Ejemplo 14.14

```
\pdfpagewidth=11cm \hsize=11cm \hoffset=-1in
\pdfpageheight=1cm \vsize=1cm \voffset=-1in
\mathcode`@="7100 % Establece el carácter '@' como una letra en modo matemático
$$
  x @ y % Usando '@' como una letra en modo matemático
$$
\bye
```

$x\Gamma y$

3 `\delcode<27 bit number> (0-"7FFFFFF)`: El código delimitador determina cómo se trata un carácter después de las instrucciones `\left` o `\right` en modo matemático.

Ejemplo 14.15

```
\pdfpagewidth=11cm \hsize=11cm \hoffset=-1in
\pdfpageheight=2cm \vsize=2cm \voffset=-1in
\delcode`@="028300 % Establece el carácter '@' como un delimitador
$$
  \left@ 1 \over 2 \right@
$$
\bye
```

$\left(\frac{1}{2}\right($

4 `\sfcode integer`: Determina cómo afecta el espaciado después de un carácter.

Ejemplo 14.16

```
\pdfpagewidth=11cm \hsize=11cm \hoffset=-1in
\pdfpageheight=1cm \vsize=1cm \voffset=-1in
\sfcode`.=3000 % Aumenta el espacio después de un punto
Texto de ejemplo. Esto muestra el espacio adicional después de un punto.
\bye
```

Texto de ejemplo. Esto muestra el espacio adicional despus de un punto.

5 `\lccode`, `\uccode<8 bit number> (0-255)`: Estos códigos se utilizan para definir las versiones en minúsculas y mayúsculas de los caracteres. Son útiles para funciones como `\lowercase` y `\uppercase`.

Ejemplo 14.17

```
\pdfpagewidth=11cm \hsize=11cm \hoffset=-1in
\pdfpageheight=1cm \vsize=1cm \voffset=-1in
\lccode`@=`a % Establece la versión en minúscula de '@' como 'a'
\uccode`@=`A % Establece la versión en mayúscula de '@' como 'A'
\lowercase{@}migo % Esto se convierte en 'amigo'
\uppercase{@}migo % Esto se convierte en 'Amigo'
\bye
```

amigo Amigo

Conversión de tokens a cadenas

El comando `\string` en TeX toma el siguiente token y lo convierte en una cadena de caracteres separados.

Ejemplo 14.18 (string como una secuencia de control)

```
\pdfpagewidth=11cm \hsize=11cm \hoffset=-1in
\pdfpageheight=1cm \vsize=1cm \voffset=-1in
\tt\string\TeX % Esto se convierte en '\TeX'
\bye
```

```
\TeX
```

Ejemplo 14.19 (string con un caracter especial)

```
\pdfpagewidth=11cm \hsize=11cm \hoffset=-1in
\pdfpageheight=1cm \vsize=1cm \voffset=-1in
\tt\string&% Esto imprimirá: &
\bye
```

```
&
```

Ejemplo 14.20 (string con signo de porcentaje)

```
\pdfpagewidth=11cm \hsize=11cm \hoffset=-1in
\pdfpageheight=1cm \vsize=1cm \voffset=-1in
\tt\string% % Esto no funciona
Esto no funciona
\bye
```

```
Esto no funciona
```

En este caso, `\string` no puede tomar el signo de porcentaje (%) como token, ya que TEX lo procesa como un comentario y lo elimina antes de que `\string` pueda actuar sobre él. Entonces, `\string` tomará el primer token en la línea siguiente y lo convertirá en una cadena de caracteres.

Una forma de lograr que el signo de porcentaje aparezca en la salida es usar una secuencia de control como %, que es independiente del mecanismo de comentarios:

Ejemplo 14.21

```
\pdfpagewidth=11cm \hsize=11cm \hoffset=-1in
\pdfpageheight=1cm \vsize=1cm \voffset=-1in
....
\tt\string\%
....
\bye
```

```
.... \% ....
```

Salida de secuencia de control

En esta sección, se discute cómo se muestran las secuencias de control en la salida de TeX y cómo se pueden modificar utilizando \escapechar. A continuación, se presentan ejemplos ejecutables para ilustrar estos conceptos.

\escapechar

1 \escapechar. El carácter de escape (\escapechar) es el carácter que precede a una secuencia de control cuando se muestra en la salida. El valor predeterminado es 92, que corresponde a la barra invertida. Si \escapechar es negativo o mayor que 255, no se muestra el carácter de escape.

Ejemplo 14.22

```
\pdfpagewidth=11cm \hsize=11cm \hoffset=-1in
\pdfpageheight=1cm \vsize=1cm \voffset=-1in
\escapechar=`/ % Cambiar el carácter de escape a '/'
\tt\string\TeX % Imprimirá: /TeX
\bye
```

/TeX

\write, \message y \errmessage

2 \write, \message y \errmessage. El valor de \escapechar también se utiliza al mostrar secuencias de control en la salida de \write, \message y \errmessage.

```
\pdfpagewidth=11cm \hsize=11cm \hoffset=-1in
\pdfpageheight=1cm \vsize=1cm \voffset=-1in
\message{Ejemplo de mensaje: \string\TeX} % Imprimirá en el
↪ registro: Ejemplo de mensaje: \TeX
No hay contenido porque message imprime en el archivo de extensión
↪ .log
\bye
```

3 \show, \showthe y \meaning.

```
\pdfpagewidth=11cm \hsize=11cm \hoffset=-1in
\pdfpageheight=1cm \vsize=1cm \voffset=-1in
\show\TeX % Imprimirá en el registro: \TeX=macro: ->\leavevmode
↪ \kern .1667em\lower .5ex\hbox {T}\kern -.125em\lower
↪ .0001ex\hbox {E}\kern -.0833em\lower .5ex\hbox {X}
No hay contenido de salida en pdf porque show imprime en el archivo
↪ de extensión .log
\bye
```

En este ejemplo, se muestra la definición de la secuencia de control \TeX en el registro utilizando el carácter de escape predeterminado.

4 \mintinline{tex}{\string}, \mintinline{tex}{\noexpand} en \mintinline{tex}{\write}. En algunas circunstancias, se puede reemplazar \string por \noexpand al utilizar \write.

```
\pdfpagewidth=11cm \hsize=11cm \hoffset=-1in
\pdfpageheight=1cm \vsize=1cm \voffset=-1in
\newwrite\miarchivo
\immediate\openout\miarchivo=ejemplo.txt
\immediate\write\miarchivo{\noexpand\TeX} % Escribirá \TeX en el
↪ archivo "ejemplo.txt" sin expansión
\immediate\closeout\miarchivo
No hay contenido de salida en pdf porque write imprime en un archivo
↪ externo.
\bye
```

Código de categoria de un string

El comando \string en TeX se utiliza para obtener la representación literal de un token en forma de una cadena de caracteres. El resultado de un comando \string es una cadena de caracteres con categoría 12, a excepción de los espacios en una secuencia de control, los cuales tienen categoría 10.

Ejemplo 14.23

```
\def\miMacro{hola mundo}
\string\miMacro % Imprimirá "hola mundo"
```

\miMacro

Al ejecutar \string\miMacro, el resultado será la cadena "hola mundo" con todos los caracteres con categoría 12, excepto el espacio que tendrá categoría 10.

Otro comando que tiene un comportamiento similar a \string es \meaning, el cual muestra la definición de una secuencia de control. Por ejemplo,

Ejemplo 14.24

```
\def\miOtraMacro#1{\textbf{#1}}
\meaning\miOtraMacro % Imprimirá: macro:#1->\textbf {#1}
```

macro:#1->\textbf {#1}

Al ejecutar \meaning\miOtraMacro, el resultado será la cadena de caracteres "\textbf{#1}", donde el símbolo # tendrá categoría 12 y el resto de los caracteres tendrán categoría 12.

14.6 Fuentes

Para usar una fuente en TEX, primero debe declararse mediante el comando \font. Este comando establece una secuencia de control que identifica la fuente y permite que TEX la use en el documento. Además de la identificación, una fuente también tiene un nombre externo que se puede mostrar en el documento mediante el comando \fontname.

En caso de que TEX no pueda encontrar una fuente, usa una fuente vacía llamada \nullfont. Esta fuente vacía es una fuente de emergencia que se usa solo si TEX no puede encontrar la fuente solicitada.

Resumen de comandos a estudiar para el modo texto:

1. \font: Crea una secuencia de control para una fuente.
2. \fontname: Muestra el nombre externo de una fuente.
3. \hyphenchar: Establece el número de carácter de guion en una fuente.
4. \fontdimen: Accede a los parámetros de una fuente.
5. \char47: Agrega correcciones itálicas a un carácter en una fuente.
6. \noboundary: Omite el carácter de límite implícito en una fuente.

Ejemplo 14.25 (Font)

```
\pdfpagewidth=11cm \hsize=11cm \hoffset=-1in
\pdfpageheight=1cm \vsize=1cm \voffset=-1in
\font\myfont=cmr10
\myfont Hola mundo!
\bye
```

Hola mundo!

En este ejemplo se declara una fuente con la secuencia de control \myfont y carga la fuente cmr10 (Computer Modern Roman, tamaño 10).

Ejemplo 14.26 (fontname)

```
\pdfpagewidth=11cm \hsize=11cm \hoffset=-1in
\pdfpageheight=1cm \vsize=1cm \voffset=-1in
\font\myfont=cmr10
El nombre externo de la fuente es: \fontname\myfont.
\bye
```

El nombre externo de la fuente es: cmr10.

Ejemplo 14.27 (nullfont)

```
\pdfpagewidth=11cm \hsize=11cm \hoffset=-1in
\pdfpageheight=1cm \vsize=1cm \voffset=-1in
...{\nullfont Este texto no será visible en el documento}...

Texto normal después del texto oculto.
\bye
```

......
Texto normal despus del texto oculto.

Ejemplo 14.28 (hyphenchar)

```
\pdfpagewidth=11cm \hsize=11cm \hoffset=-1in
\pdfpageheight=1cm \vsize=1cm \voffset=-1in
El valor predeterminado de hyphenchar es: \the\hyphenchar\font. % muestra el valor predeterminado

\font\myfont=ptmr8t
\hyphenchar\myfont=45 % establece el valor de \hyphenchar en 45
\myfont Hola-mundo.
\bye
```

El valor predeterminado de hyphenchar es: 45.
Hola-mundo.

Este ejemplo establece el valor de `\hyphenchar` en 45 y carga la fuente "ptmr8t" (Times Roman, tamaño 8). También muestra el valor predeterminado de `\hyphenchar` antes de cargar la fuente.

Ejemplo 14.29 (fontdimen)

```
\pdfpagewidth=11cm \hsize=11cm \hoffset=-1in
\pdfpageheight=1cm \vsize=1cm \voffset=-1in
Hola mundo

\font\myfont=cmr10
\fontdimen2\myfont=2em % ajusta el espacio entre letras
\myfont Hola mundo!
\bye
```

Hola mundo
Hola mundo!

Ejemplo 14.30 (char47)

```
\pdfpagewidth=11cm \hsize=11cm \hoffset=-1in
\pdfpageheight=1cm \vsize=1cm \voffset=-1in
\font\myfont=cmr10
\myfont a\char47 b % agrega una corrección itálica entre a y b
\bye
```

a/b

Ejemplo 14.31 (noboundary)

```
\pdfpagewidth=11cm \hsize=11cm \hoffset=-1in
\pdfpageheight=1cm \vsize=1cm \voffset=-1in
\font\myfont=cmr10
\noboundary\myfont Hola.
\bye
```

Hola.

Fuentes

Una fuente es el conjunto de caracteres que se encuentra en un archivo externo de fuente. Durante el procesamiento, TEX decide de qué fuente tomar un carácter, separando esta decisión para el modo de texto y el modo matemático.

En el modo de texto, los caracteres se toman de la "fuente actual", que se establece mediante una declaración de fuente con el comando `\font`. El nombre del archivo externo de la fuente se asocia a una secuencia de control. El cambio a la fuente descrita en ese archivo se realiza mediante el uso de esa secuencia de control. Las fuentes son archivos de extensión .tfm.

En el modo matemático, se usa la "familia actual", que puede contener tres fuentes diferentes.

El comando `\the\font` se utiliza para mostrar la fuente actual. Las declaraciones de fuente son locales a un grupo y se especifican mediante un archivo tfm.

La declaración de fuente tiene la sintaxis

```
\font⟨control sequence⟩⟨equals⟩⟨file name⟩⟨at clause⟩
```

, donde

```
⟨at clause⟩ → at ⟨dimen⟩ | scaled ⟨number⟩ | ⟨optional spaces⟩
```

Ejemplo 14.32

```
\pdfpagewidth=11cm \hsize=11cm \hoffset=-1in
\pdfpageheight=3cm \vsize=3cm \voffset=-1in
\font\myfont=cmr10 at 13pt
\myfont Hola mundo!

\font\MyFont=cmr10 at 7pt % Declaración de la nueva fuente

{\MyFont Esta es una frase en la nueva fuente.} % Cambio a la nueva fuente

% Volver a la fuente predeterminada
```

```
Esta es una frase en la fuente predeterminada.
\bye
```

Hola mundo!
Esta es una frase en la nueva fuente.
Esta es una frase en la fuente predeterminada.

La fuente externa necesaria es un archivo tfm, que se carga independientemente de cualquier at clause en la declaración de fuente. Si el archivo tfm ya se ha cargado, se puede reasignar sin necesidad de recurrir al archivo tfm.

Información de fuente

Cuando se ejecuta el sistema TEX busca información acerca de la fuente, como las dimensiones de los caracteres. Esta información se encuentra almacenado en archivos de extensión .tfm.

Dimensiones de fuente

Los parámetros de una fuente de texto tienen al menos 7 parámetros y para el modo matemático se tiene más parámetros. El sistema TEX toma los parámetros no especificados con un valor cero.

Los parámetros de una fuente de texto son:

1. La inclinación por punto; esta dimensión se utiliza para la posición horizontal adecuada de los acentos.
2. El espacio entre palabras: se utiliza a menos que el usuario especifique un `\spaceskip` explícito.
3. La elongación entre palabras: la componente de elongación del espacio entre palabras.
4. La compresión entre palabras: la componente de compresión del espacio entre palabras.
5. La altura x: el valor de la unidad interna ex, que generalmente es aproximadamente la altura de la letra minúscula "x".
6. La anchura de M: el valor de la unidad interna em, que es aproximadamente la anchura de la letra mayúscula "M".
7. El espacio adicional: el espacio agregado al espacio entre palabras al final de las oraciones (es decir, cuando `\spacefactor` mayor o igual que 2000) a menos que el usuario especifique un `\xspaceskip` explícito.

La modificación o la asignación global de una dimensión de fuente se realiza mediante el comando `\fontdimen`:

\fontdimen⟨number⟩⟨font⟩⟨equals⟩⟨dimen⟩

Ejemplo 14.33

```
\pdfpagewidth=11cm \hsize=11cm \hoffset=-1in
\pdfpageheight=3cm \vsize=3cm \voffset=-1in
\font\myfont=cmr10 % Definir una nueva fuente
\myfont % Cambiar a la nueva fuente
\the\fontdimen2\font

\fontdimen2\font=2\fontdimen2\font
\the\fontdimen2\font

Esto es un ejemplo de texto con espaciado de palabras modificado.
\bye
```

3.33333pt
6.66666pt
Esto es un ejemplo de texto con espaciado de palabras modificado.

14.7 Cajas

Esta sección trata sobre los conceptos y comandos en TEXrelacionados con los "cajas" o "box registers" . Una "caja" es un contenedor para listas horizontales o verticales de elementos. Se pueden almacenar cajas en "box registers". Los conceptos tratados incluyen las dimensiones de las cajas, la forma en que se colocan sus componentes entre sí y cómo se pueden utilizar los "box registers".

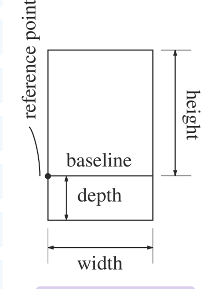


Figura 14.1

Caja

Los siguientes son algunos de los comandos que se mencionan:

1. \hbox: Este comando crea una caja horizontal que contiene texto y/o comandos. La caja horizontal tiene una anchura determinada por su contenido, mientras que su altura y profundidad son determinadas por las dimensiones de las letras y símbolos dentro de la caja. Las cajas horizontales son útiles para controlar el espaciado y el formato del texto.

2. \vbox: Este comando crea una caja vertical con el punto de referencia en el último elemento. La caja vertical almacena contenido en una columna y se utiliza para organizar el contenido en un flujo vertical en lugar de uno horizontal. La altura, anchura y profundidad de la caja son determinadas por su contenido.

3. \vtop: Similar a \vbox, este comando crea una caja vertical, pero su punto de referencia se encuentra en el primer elemento. Esto puede ser útil cuando se desea alinear varias cajas verticales por su primer elemento en lugar de su último elemento.

4 `\vcenter`: Este comando crea una caja vertical centrada en el eje matemático. Es útil para alinear elementos en un entorno matemático, como alinear una expresión en un sistema de ecuaciones. Este comando sólo puede ser utilizado en modo matemático.

5 `\vsplit`: Este comando divide la parte superior de una caja vertical. Se utiliza para extraer contenido de una caja vertical y colocarlo en otra caja. Es útil para manejar cajas grandes y dividirlas en partes más pequeñas.

6 `\box`: Este comando usa un "box register", vaciándolo. Los "box registers" son almacenamientos temporales para cajas. `\box` se utiliza para extraer el contenido de un "box register" y colocarlo en la lista actual de comandos.

7 `\setbox`: Este comando asigna una caja a un "box register". Se utiliza para guardar cajas y manipular su contenido antes de insertarlas en la lista de comandos.

8 `\copy`: Este comando funciona de manera similar a `\box`, pero no vacía el "box register". Esto permite que el contenido del "box register" se utilice varias veces.

9 `\ifhbox / \ifvbox`: Estos comandos verifican si un "box register" contiene una caja horizontal o vertical, respectivamente. Esto es útil para condicionales que dependen del tipo de caja almacenada en un "box register".

10 `\ifvoid`: Este comando verifica si un "box register" está vacío. Es útil para condicionales que requieren saber si un "box register" contiene contenido o no.

11 `\newbox`: Este comando asigna un nuevo "box register" para almacenar una caja. Es útil para crear múltiples registros de cajas para uso en diversas partes del documento.

12 `\lastbox`: Este comando devuelve la última caja de la lista actual de comandos. Es útil para extraer la última caja de una lista de comandos y colocarla en un "box register".

13 `\ht / \dp / \wd`: Estos comandos devuelven las dimensiones de una caja almacenada en un "box register": altura (`\ht`), profundidad (`\dp`) y anchura (`\wd`). Son útiles para calcular y ajustar el tamaño de las cajas y su contenido.

14 `\boxmaxdepth`: Este comando establece la profundidad máxima permitida para las cajas. Por defecto en TEX, su valor es `\maxdimen`. Este parámetro se puede modificar para limitar la profundidad de las cajas, evitando que sean demasiado grandes.

15 `\splitmaxdepth`: Este comando establece la profundidad máxima permitida para las cajas generadas por `\vsplit`. Es útil para limitar el tamaño de las cajas resultantes después de dividir una caja vertical.

16 `\badness`: Este comando devuelve el grado de "malformación" de la caja más recientemente construida. Cuanto mayor sea el valor de `\badness`, peor será la calidad del ajuste del contenido de la caja. Este parámetro es útil para identificar y ajustar el formato de las cajas en función de su calidad.

17 `\hfuzz` / `\vfuzz`: Estos comandos establecen el tamaño excesivo que TEXtolera antes de considerar una caja horizontal (`\hfuzz`) o vertical (`\vfuzz`) como "sobresaliente". Son útiles para controlar la precisión con la que se ajusta el contenido de las cajas.

18 `\hbadness` / `\vbadness`: Estos comandos establecen la cantidad de tolerancia antes de que TEXinforme una caja horizontal o vertical como "inferior" o "sobresaliente". Cuando la "malformación" de una caja, medida por el parámetro `\badness`, supera el valor establecido por `\hbadness` o `\vbadness`, TEXgenera un mensaje de advertencia. Estos comandos son útiles para controlar cuándo TEXemite advertencias sobre la calidad del ajuste en las cajas y permiten al usuario ajustar el nivel de tolerancia según sus necesidades.

19 `\overfullrule`: Este comando establece el ancho de la regla que se imprime para indicar cajas horizontales "sobresalientes". Cuando una caja horizontal es demasiado ancha y supera el tamaño permitido por `\hfuzz`, TEXpuede marcarla visualmente dibujando una regla negra junto a la caja. El ancho de esta regla se controla con `\overfullrule`, lo que permite al usuario ajustar su apariencia.

20 `\hsize`: Este comando define el ancho de línea utilizado para el tipo de texto dentro de una caja vertical. Afecta cómo se ajusta el texto dentro de la caja, y se usa comúnmente para controlar el ancho de las columnas de texto, como en un diseño de varias columnas.

21 `\vsize`: Este comando define la altura de la caja de página, que es el área en la que TEXcoloca el contenido al componer una página. Ajustar este valor afecta la cantidad de contenido que cabe en una página y, en consecuencia, el número de páginas en un documento.

22 `\lastbox`: Este registro contiene el último elemento agregado a la lista actual, si este fue una caja. Es útil cuando se necesita referenciar o manipular la última caja agregada a una lista horizontal o vertical.

23 `\raise` y `\lower`: Estos comandos ajustan la posición vertical de una caja en modo horizontal. Permiten modificar la alineación vertical de elementos en una lista horizontal, como cuando se desea desplazar una imagen o una línea de texto hacia arriba o hacia abajo. Por otro lado, `\moveleft` y `\moveright` ajustan la posición horizontal de una caja en modo vertical, lo que permite modificar la alineación horizontal de elementos en una lista vertical.

24 `\everyhbox` y `\everyvbox`: Estos comandos son listas de tokens que se insertan al inicio de una caja horizontal y vertical, respectivamente. Permiten especificar acciones específicas que se deben realizar en cada caja que se crea. Por ejemplo, se pueden usar para insertar un espacio uniforme entre todas las cajas horizontales o verticales, aplicar un estilo común a todas las cajas, o insertar automáticamente ciertos elementos, como números de línea o marcas de agua.

Nota: algunos de estos macros ya se han desarrollado, por lo tanto no se desarrollarán tales macros.

Cajas

Cuando TEXespera una ⟨caja⟩, cualquiera de las siguientes formas es admisible:

- \hbox<box specification><horizontal material>
- \vbox<box specification><vertical material>
- \vtop<box specification><vertical material>
- \box<8-bit number>
- \copy<8-bit number>
- \vsplit<8-bit number>to<dimen>
- \lastbox

Una <box specification> se define como:

```
<box specification> → <filler>
| to <dimen><filler> | spread <dimen><filler>
```

Un <8-bit number> es un número en el rango 0-255.

Una <box> en general se puede usar en modo horizontal, vertical y matemático.

La operación \vsplit se trata a detalle en la sección 14.21.

Registro de cajas

Existen 256 registros de cajas en TEX, las cuales son numeradas del 0 a 255. Cada registro puede contener una caja horizontal o vertical.

newbox

El macro \newbox asigna una caja nueva a un registro de caja.

setbox, box y copy

Sintaxis de \setbox: \setbox<number><equals><box>. Ejemplo:

```
\setbox0=\hbox{Hola mundo}
\setbox\mybox\hbox{Hola mundo}
```

Ejemplo 14.34 (copy)

```
\pdfpagewidth=11cm \hsize=11cm \hoffset=-1in
\pdfpageheight=1cm \vsize=1cm \voffset=-1in
\newbox\mybox
\setbox\mybox=\vbox{Hola mundo} % Asignamos contenido al registro mybox
```

```
\noindent\copy\mybox\par
\noindent\copy\mybox
\bye
```

Hola mundo
Hola mundo

Ejemplo 14.35 (vsplit)

```
\pdfpagewidth=11cm \hsize=11cm \hoffset=-1in
\pdfpageheight=4cm \vsize=4cm \voffset=-1in
\newbox\originalbox
\newbox\splitbox
\setbox\originalbox=\vbox{%
  Hola mundo Hola mundo Hola mundo Hola mundo Hola mundo Hola mundo
  Hola mundo Hola mundo Hola mundo Hola mundo Hola mundo Hola mundo
  Hola mundo Hola mundo Hola mundo Hola mundo Hola mundo Hola mundo
  Hola mundo Hola mundo Hola mundo Hola mundo Hola mundo Hola mundo
} % Asignamos contenido al registro originalbox
% Dividimos originalbox en splitbox con una altura máxima de 1cm
\setbox\splitbox=\vsplit\originalbox to 1cm
% Mostramos el contenido de splitbox y originalbox
\noindent Split box:\par
\noindent\unvbox\splitbox
\bigskip
\noindent Contenido de restante de la caja original:\par
\noindent\unvbox\originalbox
\bye
```

Split box:
Hola mundo Hola mundo Hola mundo Hola mundo Hola mundo
Hola mundo Hola mundo Hola mundo Hola mundo Hola mundo Hola

Contenido de restante de la caja original:
mundo Hola mundo Hola mundo Hola mundo Hola mundo Hola mundo
Hola mundo Hola mundo Hola mundo Hola mundo Hola mundo Hola
mundo Hola mundo Hola mundo

ifvoid, ifhbox y ifvbox

El macro `\ifvoid<number>` se utiliza para verificar si un registro de caja está vacío.

Los macros `\ifhbox<number>` y `\ifvbox<number>` se utilizan para verificar si un registro de caja contiene una caja horizontal o vertical, respectivamente.

lastbox

`\lastbox` se comporta como un registro de caja. Si el último elemento de la lista actual no es una caja, entonces `\lastbox` es un caja vacía.

El código: `{\setbox0=\lastbox}`. Elimina la última caja de la lista actual y la asigna al registro de caja 0. Cuando la asignación ocurre dentro de un grupo, en este caso las llaves, el registro se elimina al final del grupo, en este caso las llaves.

Estudiaremos más detalle el comando `\lastbox` en la sección de rutina de salida.

Cuando se utilicen los macros como `\moveright`, `\moveleft`, `\raise` y `\lower` para desplazar la última caja, esta propiedad no se hereda cuando se utiliza `\lastbox`.

Dimensión natural de una caja

Dimensiones de las cajas horizontales

Para desplazar una caja horizontal con referencia al punto de baseline se debe de utilizar los comandos `\lower` y `\raise`.

Las cajas horizontales se alinean uno al otro lado, en dónde sus líneas de base se encuentran en una sola línea.

Ejemplo 14.36 (Desplazamiento vertical)

```
\pdfpagewidth=11cm \hsize=11cm \hoffset=-1in
\pdfpageheight=3cm \vsize=3cm \voffset=-1in
Hola \lower1cm\hbox{mundo}...

Hola \raise1cm\hbox{mundo}...
\bye
```

Hola ...

mundo
mundo

Hola ...

Ejemplo 14.37 (Ancho positivo y negativo de una caja)

```
\pdfpagewidth=11cm \hsize=11cm \hoffset=-1in
\pdfpageheight=1cm \vsize=1cm \voffset=-1in
Ancho positivo: a\hbox{\hskip2cm b}\par
Ancho negativo: a\hbox{\hskip-2cm b}
\bye
```

Ancho positivo: a b
Ancho negativo: a b

Dimensiones de las cajas verticales

El material dentro de una caja vertical se alinea con referencia al línea vertical que pasa por el punto de referencia de la caja.

Una caja vertical se puede mover con los comandos `\moveleft` y `\moveright`.

Ejemplo 14.38 (Desplazamiento horizontal)

```
\pdfpagewidth=11cm \hsize=11cm \hoffset=-1in
\pdfpageheight=1cm \vsize=1cm \voffset=-1in
\moveleft1cm\vbox{Hola mundo 1}\par
\moveright1cm\vbox{Hola mundo 2}
\bye
```

ola mundo 1
Hola mundo 2

El ancho de una caja vertical es igual al ancho del material que se contiene en la caja. El contenido de la caja se adhiere a la derecha del punto de referencia. Por ejemplo,

Ejemplo 14.39

```
\pdfpagewidth=11cm \hsize=10cm \hoffset=-1in
\pdfpageheight=1cm \vsize=1cm \voffset=-1in
a\vbox{\hbox{\kern-2em b}}...
\bye
```

b a...

En una caja `\vbox` su punto de referencia se hubica en la línea de base y de `\vtop` se hubica en la linea base del primer componente.

Ejemplo 14.40

```
\pdfpagewidth=11cm \hsize=11cm \hoffset=-1in
\pdfpageheight=3cm \vsize=3cm \voffset=-1in
\parindent=0pt
aaaaa \vbox{\hsize 2cm Hola mundo, hola mundo, hola mundo}---

bbbbb \vtop{\hsize 2cm Hola mundo, hola mundo, hola mundo}...
\bye
```

```
        Hola mundo,
        hola mundo, hola█
aaaaa mundo        —
bbbbb Hola mundo,...
         hola mundo, hola█
         mundo
```

Cuando la profundidad de cajas verticales excede a `\boxmaxdepth`, entonces la caja se mueve hacia abajo en la cantidad excedente con referencia al punto de referencia de la caja.

Ejemplos

Ejemplo 14.41 (Cajas horizontales)

```
\pdfpagewidth=11cm \hsize=11cm \hoffset=-1in
\pdfpageheight=1cm \vsize=1cm \voffset=-1in
\parindent=0pt
\setbox0=\hbox{aa}
\setbox1=\hbox{\copy0 \hskip-\wd0}
Ancho de caja 1: \the\wd1 % muestra la anchura de la caja 1, debería ser 0
\bye
```

Ancho de caja 1: 0.0pt

Ejemplo 14.42 (Altura y profundidad de cajas horizontale)

```
\pdfpagewidth=11cm \hsize=11cm \hoffset=-1in
\pdfpageheight=1cm \vsize=1cm \voffset=-1in
\parindent=0pt
\setbox0=\hbox{\vrule height 5pt depth 5pt}
\setbox1=\hbox{\raise 10pt \box0}
Altura de caja 1: \the\ht1 \par % muestra la altura de la caja 1, debería ser 15pt
Profundidad de caja 1: \the\dp1 % muestra la profundidad de la caja 1, debería ser 0pt
\bye
```

Altura de caja 1: 15.0pt
Profundidad de caja 1: 0.0pt

Ejemplo 14.43 (Cajas verticales)

```
\pdfpagewidth=11cm \hsize=11cm \hoffset=-1in
\pdfpageheight=1cm \vsize=1cm \voffset=-1in
\parindent=0pt
\setbox0=\hbox{\hskip 10pt}
\setbox1=\vbox{\moveleft 5pt \copy0}
\setbox2=\vbox{\moveright 5pt \copy0}
```

```
Ancho de caja 1: \the\wd1 \par% muestra la anchura de la caja 1, debería ser 5pt
Ancho de caja 2: \the\wd2 % muestra la anchura de la caja 2, debería ser 15pt
\bye
```

Ancho de caja 1: 5.0pt
Ancho de caja 2: 15.0pt

Ejemplo 14.44 (Profundidad de una caja vertical)

```
\pdfpagewidth=11cm \hsize=11cm \hoffset=-1in \pdfpageheight=2cm \vsize=2cm \voffset=-1in
\parindent=0pt
\setbox0=\vbox{\vskip 5pt \hbox{\vrule height 5pt depth 5pt}}
\setbox1=\vbox{\vskip -5pt \hbox{\vrule height 5pt depth 5pt}}
Altura de caja 0: \the\ht0\par % muestra la altura de la caja 0, debería ser 10pt
Profundidad de caja 0: \the\dp0\par % muestra la profundidad de la caja 0, debería ser 5pt
Altura de caja 1 : \the\ht1 \par % muestra la altura de la caja 1, debería ser 0pt
Profundidad de caja 1: \the\dp1 % muestra la profundidad de la caja 1, debería ser 5pt
\bye
```

Altura de caja 0: 10.0pt
Profundidad de caja 0: 5.0pt
Altura de caja 1 : 0.0pt
Profundidad de caja 1: 5.0pt

Ejemplo 14.45 (Cajas verticales con vtop)

```
\pdfpagewidth=11cm \hsize=11cm \hoffset=-1in
\pdfpageheight=2cm \vsize=2cm \voffset=-1in
\parindent=0pt
\setbox0=\vtop{\vskip 5pt \hbox{\vrule height 5pt depth 5pt}}
\setbox1=\vtop{\vskip -5pt \hbox{\vrule height 5pt depth 5pt}}
Altura de caja 0: \the\ht0 \par% muestra la altura de la caja 0, debería ser 0pt
Profundidad de caja 0: \the\dp0 \par% muestra la profundidad de la caja 0, debería ser 15pt
Altura de caja 1: \the\ht1 \par% muestra la altura de la caja 1, debería ser 0pt
Profundidad de caja 1: \the\dp1 % muestra la profundidad de la caja 1, debería ser 5pt
\bye
```

Altura de caja 0: 0.0pt
Profundidad de caja 0: 15.0pt
Altura de caja 1: 0.0pt
Profundidad de caja 1: 5.0pt

Más información acerca de las dimensiones de una caja

Dimensiones predeterminadas

En TeX, cuando se crea una caja, es posible especificar su tamaño de antemano utilizando una ⟨especificación de caja⟩. Esto se hace con las siguientes instrucciones:

```
\hbox to ⟨dimen⟩ {...}    \vbox to ⟨dimen⟩ {...}
```

Aquí, `<dimen>` es una dimensión que especificas (como 5cm o 10pt), y `{...}` es el contenido de la caja. El `to <dimen>` indica a TEX que debe hacer que la caja tenga ese tamaño.

Por ejemplo, si quieres crear una caja horizontal de 5cm con el texto "Hola Mundo", se realizaría la siguiente operación:

```
\hbox to 5cm {Hola Mundo}
```

Si existe "glue" en la caja que puede estirarse o encogerse, TeX ajustará ese espacio para que la caja alcance el tamaño especificado. Este ajuste de glue se asocia con un valor de "badness" (maldad), que es un indicador de qué tan forzado fue este ajuste.

Si no hay suficiente "glue" para estirar o encoger, según sea necesario, la caja resultante estará "underfull" (infra-llena,incompletamente lleno, subocupado) o "overfull" (sobre-llena, sobrellenado), respectivamente. Estos casos generalmente generan advertencias o errores en TeX.

Ejemplo 14.46 (Texto con suficiente "glue")

```
\pdfpagewidth=11cm \hsize=11cm \hoffset=-1in \pdfpageheight=1cm \vsize=1cm \voffset=-1in
\parindent=0pt
\hbox {Este texto tiene suficiente glue.}
\bye
```

Este texto tiene suficiente glue.

Ejemplo 14.47 (Caja de texto con muy poco "glue" (caja underfull))

```
\pdfpagewidth=11cm \hsize=11cm \hoffset=-1in \pdfpageheight=1cm \vsize=1cm \voffset=-1in
\parindent=0pt
\hbox to 100pt{Texto}
\bye
```

Texto

Ejemplo 14.48 (Caja de texto sin "glue" (caja overfull))

```
\pdfpagewidth=11cm \hsize=11cm \hoffset=-1in \pdfpageheight=1cm \vsize=1cm \voffset=-1in
\parindent=0pt
```

```
\hbox to 50pt{Este texto demasiado largo para la caja.}
\bye
```

Este texto demasiado largo para la caja.

Además, existe otra instrucción que permite que una caja tenga un tamaño distinto al tamaño natural, que se realiza con spread ⟨dimen⟩:

```
\hbox spread ⟨dimen⟩ {...}    \vbox spread ⟨dimen⟩ {...}
```

Esta instrucción indica a TeX que ajuste el glue de tal manera que el tamaño de la caja sea una cantidad específica mayor que el tamaño natural. El uso de estos comandos ya se estudiaron en el capítulo 13.

Cambio de dimensiones de una caja

Las dimensiones de una caja en TeX (altura, profundidad y ancho) son accesibles a través de los comandos `\ht`, `\dp` y `\wd` respectivamente. Por ejemplo, `\dp10` proporcionaría la profundidad de la caja 10.

No solamente podemos obtener las dimensiones de una caja, también podemos establecer los valores o las dimensiones de estas cajas a través de los comandos ya mencionados. Al realizar esta operación no se cambia las dimensiones del contenido de la caja, es decir no cambia la forma en que el pegamento (glue) se ajusta.

Ejemplo 14.49 (Altura y profundidad eliminado en una caja)

```
\pdfpagewidth=11cm \hsize=11cm \hoffset=-1in \pdfpageheight=1cm \vsize=1cm \voffset=-1in
\parindent=0pt
\def\smash#1{{\setbox0=\hbox{#1}\dp0=0pt \ht0=0pt \box0\relax}}
\smash{Este texto esta en una caja de altura y profundidad cero.}
\bye
```

Este texto esta en una caja de altura y profundidad cero.

Ejemplo 14.50

```
\pdfpagewidth=11cm \hsize=11cm \hoffset=-1in \pdfpageheight=1cm \vsize=1cm \voffset=-1in
\parindent=0pt
\setbox0=\hbox{---} {\wd0=0pt} a\box0b
\bye
```

ab—

Moviendo cajas

Utlizar los comandos:

```
\raise ⟨dimen⟩ ⟨box⟩          \lower ⟨dimen⟩ ⟨box⟩
\moveleft ⟨dimen⟩ ⟨box⟩       \moveright ⟨dimen⟩ ⟨box⟩
```

Dimensión y ubicación de la caja

En TEX, cuando estás construyendo listas horizontales y verticales (básicamente agrupaciones de elementos como caracteres, glifos, otras cajas, etc.), TeX utiliza un sistema de referencia y de posicionamiento.

Para las listas horizontales, piensa en la línea de referencia como la línea base en la que se asientan los caracteres de un texto. Al agregar elementos a esta lista, TeX sigue ciertos pasos:

1. Comienza desde el punto de referencia de la caja que contiene la lista.
2. Si el elemento que se va a agregar ha sido movido por comandos como `\raise` o `\lower`, ajusta la posición actual acordemente.
3. Si el elemento es una caja (horizontal o vertical), coloca su contenido de manera recursiva usando el mismo procedimiento.
4. Finalmente, mueve la posición actual a la derecha por el ancho del componente que acabas de agregar.

Para las listas verticales, TeX mantiene una línea de referencia vertical. Aquí, la posición actual inicialmente está en la esquina superior izquierda de la caja que contiene la lista. Luego, sigue pasos similares a los de la lista horizontal pero de manera vertical. Aquí está el proceso:

1. Si un elemento ha sido desplazado por `\moveleft` o `\moveright`, TeX ajusta la posición actual.
2. Coloca el componente con su esquina superior izquierda en la posición actual.
3. Si el componente es una caja (horizontal o vertical), coloca su contenido de manera recursiva.
4. Finalmente, mueve la posición actual hacia abajo por la altura más la profundidad de la caja que acabas de agregar.

Estos procesos no están construyendo la caja en sí, solo están decidiendo cómo se deben colocar los componentes dentro de la caja cuando se completa.

Ejemplo 14.51

```
\pdfpagewidth=11cm \hsize=11cm \hoffset=-1in \pdfpageheight=1cm \vsize=1cm \voffset=-1in
\parindent=0pt
% Creación de una caja horizontal con texto
```

```
\setbox0=\hbox{Hola Mundo}
\the\wd0 % Muestra la anchura de la caja
\the\ht0 % Muestra la altura de la caja
\the\dp0 % Muestra la profundidad de la caja
\box0 % Imprime la caja

% Creación de una caja vertical con varias cajas horizontales (cada una con un texto)
\setbox1=\vbox{\hbox{Hola} \hbox{Mundo}}
\the\wd1 % Muestra la anchura de la caja
\the\ht1 % Muestra la altura de la caja
\the\dp1 % Muestra la profundidad de la caja
\box1 % Imprime la caja
\bye
```

54.44458pt6.94444pt0.0ptHola Mundo

Hola
30.83342pt18.94444pt0.0ptMundo

Cajas y el pegamento negativo

En ciertas ocasiones, es útil que las cajas en TeX se superpongan en lugar de alinearse una al lado de la otra. Una forma sencilla de lograr esto es utilizando "pegamento negativo" (negative glue).

Piensa en el "pegamento" como un espacio flexible que TeX puede ajustar para alinear el texto de manera óptima. En general, este espacio puede estirarse o encogerse según sea necesario. Sin embargo, también es posible utilizar un pegamento negativo para hacer que las cajas se superpongan.

Por ejemplo, puedes tener una caja (digamos, caja 8) que quieras colocar sin mover la ubicación actual. Esto se puede hacer de la siguiente manera:

```
{\dimen0=\wd8 \box8 \kern-\dimen0}
```

En este ejemplo, `\wd8` obtiene el ancho de la caja 8 y luego `\kern-\dimen0` crea un espacio negativo del mismo tamaño, lo que resulta en una superposición.

Las macros `\llap` y `\rlap` ofrecen una forma más versátil de hacer esto. Estas macros te permiten introducir material que sobresalga a la izquierda o a la derecha de la ubicación actual, respectivamente.

```
\def\llap#1{\hbox to 0pt{\hss #1}}
\def\rlap#1{\hbox to 0pt{#1\hss}}
```

`\hss` es un tipo de pegamento que absorbe cualquier ancho positivo o negativo del argumento de `\llap` o `\rlap`.

Cajas llenas (overfull) y poco llenas (underfull)

En TeX, cuando se especifica un tamaño para una caja, TeX ajusta el "pegamento" (que es esencialmente el espacio) en la caja para adaptarse a esa especificación. A veces este ajuste puede ser forzado, y este nivel de forzamiento es lo que TeX llama "badness" (maldad).

Si el "pegamento" tiene capacidades limitadas para estirarse o encogerse, se realiza un cálculo de "badness" para evaluar el grado de forzamiento de este ajuste. Los valores de "badness" pueden ir desde 0 hasta 10,000. Sin embargo, si la caja se considera excesivamente llena (overfull), la "badness" se fija en 1,000,000.

TeX emitirá un mensaje de diagnóstico cuando considere que la "badness" es excesivamente alta.

Una caja se considera sobre-llena (overfull) cuando el glue de encogimiento disponible en la caja es menor que el encogimiento necesario. TeX informa un overfull cuando la diferencia de congimiento es más de `\hfuzz` o si `\hbadness < 100`.

Ahora, en lo que respecta a las cajas horizontales, si el "pegamento" de una caja necesita estirarse, TeX no informará nada si `\hbadness` es mayor o igual a 10000. Si la "badness" es mayor que `\hbadness`, TeX lo reportará como "underfull" (insuficientemente lleno).

```
\hfuzz1pt
\hbox to 1pt{\hskip3pt minus .5pt}
\hbox to 1pt{\hskip3pt minus 1.5pt}
```

La primera caja producirá un error ya que execede en 1.5pt y la sugunda se encuentra en el límite de `\hfuzz`.

De manera similar se analiza las cajas verticales.

Abrir y cerrar cajas

El contenido de las cajas se puede delimitar por llaves o por comandos como `\bgroup` y `\egroup`. Estos dos últimos comandos no son más que que un `\let` de llave de apertura y cierre.

Después de la llave de apertura del contenido de la caja se insertan los tokens `\everyhbox` y `\everyvbox`, respectivamente. Si una caja tiene la asignación por `\setbox`, cualquier token `\afterassignment` antes de `everybox`.

Ejemplo 14.52

```
\pdfpagewidth=11cm \hsize=11cm \hoffset=-1in \pdfpageheight=1cm \vsize=1cm \voffset=-1in
\parindent=0pt
\everyhbox{hola mundo}
\afterassignment a
\setbox0=\hbox{c}
Caja sin setbox: \hbox{Heber}\par
```

```
Caja con setbox: \box0
\bye
```

Caja sin setbox: hola mundoHeber
Caja con setbox: ahola mundoc

La llaves implícitas como `\bgroup` y `\egroup` nos ayudan que a cerremos y abramos las cajas por medio de macros, como ocurre en LaTeX (específicamente la caja currbox). El siguiente código:

Ejemplo 14.53

```
\pdfpagewidth=11cm \hsize=11cm \hoffset=-1in \pdfpageheight=1cm \vsize=1cm \voffset=-1in
\hbox\bgroup Contenido de la caja \the\pageno \egroup
\bye
```

Contenido de la caja 1

Desempaquetado de cajas

Las cajas se desempaqueta con los comandos `\unhbox` y `\unvbox`. Las cajas también se pueden desempaquetar, realizando copias con los comandos `\unvcopy` y `\unhcopy`.

Cuando se desempaqueta una caja de un registro, los añade a la lista circundante. Entonces, cualquier espacio puede ser establecido de nuevo.

Ejemplo 14.54

```
\pdfpagewidth=11cm \hsize=11cm \hoffset=-1in \pdfpageheight=1cm \vsize=1cm \voffset=-1in
\parindent=0pt
\setbox0=\hbox to 1cm{-\hfil-}
a\hbox to 3cm{\unhbox0}b
\bye
```

a-   -b

Este ejemplo es equivalente a:

```
a\hbox to 3cm{-\hfil-}b
```

Textos en cajas

En una caja horizontal un texto se coloca en líneas rectas en donde el ancho es el ancho natural del texto que contiene. En una caja horizontal, no se permiten comandos verticales, por ejemplo, el comando `\par` no hace nada dentro de caja horizontal.

En una caja vertical, un texto se construye como si fuera una página, por ejemplo los párrafos se construyen en modo horizontal para luego romperse en líneas utilizando el parámetro `\hsize`.

Ejemplo 14.55

```
\pdfpagewidth=11cm \hsize=11cm \hoffset=-1in \pdfpageheight=1cm \vsize=1cm \voffset=-1in
\parindent=0pt
\hbox to 3cm{\vbox{Texto largo en una caja vertical y esto a la vez es un párrafo \par Nuevo párrafo}}
\bye
```

Texto largo en una caja vertical y esto a la vez es un prrafo
Nuevo prrafo

El caja del ejemplo es análogo al caja:

```
\vbox{\hsize=3cm Texto largo en una caja vertical y esto a la vez es un
↪ párrafo \par Nuevo párrafo}
```

Cuando un texto se encuentra dentro de una caja vertical, dicho texto se divide en líneas, las cuales se agrupan en cajas horizontales. Estas cajas se apilan en modo vertical interno, separadas posiblemente por `\baselineskip` y `\lineskip`. Este proceso también se emplea para el texto en la página, donde las cajas se apilan en modo vertical externo.

Otras observaciones sobre cajas

Cajas y contadares

Un contador en TeX es una variable que almacena un número entero. Se puede crear un nuevo contador utilizando el comando `\newcount`, y luego se puede acceder a ese contador simplemente usando su nombre. Por ejemplo, si creamos un contador llamado `\foo` con `\newcount\foo`, podemos usar `\foo` para obtener el valor del contador `\foo`.

Las cajas, por otro lado, son un poco más complicadas. Una caja en TeX es un contenedor que puede contener texto, espacios en blanco, y otras cajas. Se puede crear una nueva caja con el comando `\newbox`, pero a diferencia de los contadores, no puedes acceder directamente a una caja simplemente usando su nombre. En cambio, debes usar el comando `\box` seguido del nombre de la caja. Por ejemplo, si creamos una caja llamada `\foo` con `\newbox\foo`, debemos usar `\box\foo` para acceder a la caja `\foo`.

Cajas especiales

En TeX, algunos registros de cajas tienen propósitos especiales y se utilizan de manera interna por el sistema. Aquí hay tres ejemplos notables:

1. `\box255`: Esta caja es utilizada internamente por TeX para pasar la página a la rutina de salida. En otras palabras, cuando TeX ha decidido que una página está lista para ser enviada a la salida (es

decir, para ser mostrada o impresa), coloca el contenido de esa página en y luego lo envía a la rutina de salida.

2. `\voidb@x`: Este es el número de un registro de caja que se asigna en plain.tex, un formato base de TeX. Se supone que esta caja debe estar siempre vacía. Se usa en la macro `\leavevmode` y en otras. La macro `\leavevmode` es un comando que fuerza a TeX a entrar en el modo horizontal (el modo en el que se construyen las líneas de texto), y utiliza `\voidb@x` como parte de su implementación.

3. `\newinsert`: Cuando se crea una nueva inserción con la macro `\newinsert` en plain TeX, se reservan varios registros con el mismo número para esa inserción. En particular, se reserva un `\count`, un `\dimen`, un `\skip` y un `\box`. El conteo de estos registros comienza desde 254 y disminuye. Las inserciones son un tipo especial de cajas que se utilizan para cosas como notas al pie y flotantes.

Ejemplo 14.56

```
\pdfpagewidth=11cm \hsize=11cm \hoffset=-1in \pdfpageheight=1cm \vsize=1cm \voffset=-1in
Hola mundo
\output={\shipout\box255}
\bye
```

Hola mundo

En LaTeX el comando `\def\leavevmode{\unhbox\voidb@x}` cambia de modo vertical a modo horizontal sin agregar nada a la lista horizontal. Sin embargo, el `\indent` posterior provocado por esta transición agrega una caja de indentación. En modo horizontal, el comando `\leavevmode` no tiene efecto. Nota que aquí no es necesario usar `\unhcopy`, porque el registro ya está vacío.

Altura de caja vertical en modo horizontal

Cuando se coloca una caja vertical en modo horizontal, la referencia de alineación es el punto de referencia de la caja vertical con respecto a la caja horizontal circundante. Para entender esto, considera una caja como un recipiente que tiene una cierta altura y profundidad. Cuando TeX coloca una caja vertical en modo horizontal, comienza en la esquina superior izquierda de la caja vertical, es decir, el punto que está por encima del punto de referencia por una distancia igual a la altura de la caja.

Por lo tanto, cambiar la altura de una caja vertical cambia la posición de su contenido, porque estás moviendo la esquina superior izquierda de la caja.

Por ejemplo, considera el siguiente código:

Ejemplo 14.57

```
\pdfpagewidth=11cm \hsize=11cm \hoffset=-1in \pdfpageheight=1cm \vsize=1cm \voffset=-1in
\hbox{a\setbox0=\vbox{\hbox{b}}\box0 c}
\bye
```

abc

Ahora, si cambiamos la altura de la caja vertical a 0cm con , el resultado será el mismo visualmente, porque estamos reduciendo la altura de la caja vertical, pero el contenido "b" se mantiene en la misma posición vertical respecto al texto circundante:

Ejemplo 14.58

```
\pdfpagewidth=11cm \hsize=11cm \hoffset=-1in \pdfpageheight=1cm \vsize=1cm \voffset=-1in
\hbox{a\setbox0=\vbox{\hbox{b}}\ht0=0cm\box0 c}
\bye
```

a c
b

Colgando el lastbox de nuevo en la lista

Este fragmento describe una forma de desmontar y reconstruir un párrafo en TeX para conseguir un efecto de justificación particular. A veces, en la composición de anuncios publicitarios, se usa un estilo que se encuentra entre la justificación completa y el alineado a la derecha (flushright y raggedright). En este estilo, las líneas que se estirarían más allá de ciertos límites se ajustan con su espacio (pegamento) en ancho natural. Este proceso se puede lograr con una serie de macros en TeX, las cuales se describen a continuación.

En primer lugar, se definen dos nuevas cajas, `\linebox` y `\snapbox`, que se utilizarán para manipular cada línea del párrafo:

```
\newbox\linebox \newbox\snapbox
```

A continuación, se define una macro, `\eatlines`, que toma la última línea del párrafo (usando `\lastbox`), verifica si la línea está vacía (usando `\ifvoid`), y si no está vacía, se retiran los espacios y penalidades (penalties) al final de la línea (usando `\unskip` y `\unpenalty`), y se llama a la macro `\eatlines` recursivamente para procesar el resto del párrafo.

```
\def\eatlines{
  \setbox\linebox\lastbox
  \ifvoid\linebox
  \else
    \unskip\unpenalty
    {\eatlines}
```

Luego, la línea se "colapsa" (es decir, se ajusta) copiándola en `\snapbox` y luego desempaquetándola (usando `\unhcopy`). Se verifica si el ancho de la línea colapsada es menor que el 98% del ancho original de la línea. Si es así, se utiliza la versión colapsada de la línea; de lo contrario, se utiliza la línea original.

```
      \setbox\snapbox\hbox{\unhcopy\linebox}
      \ifdim\wd\snapbox<.98\wd\linebox
        si-\box\snapbox
      \else
        no-\box\linebox
      \fi
    \fi}
```

Finalmente, puedes usar esta macro `\eatlines` para ajustar un párrafo envolviendo el texto del párrafo en un `\vbox` y luego llamando a `\eatlines` después de `\par`, como se muestra a continuación:

Ejemplo 14.59

```
\pdfpagewidth=11cm \hsize=11cm \hoffset=-1in \pdfpageheight=3cm \vsize=3cm \voffset=-1in
\input plipsum
\newbox\linebox \newbox\snapbox
\def\eatlines{
  \setbox\linebox\lastbox
  \ifvoid\linebox
  \else
    \unskip\unpenalty
    {\eatlines}
    \setbox\snapbox\hbox{\unhcopy\linebox}
    \ifdim\wd\snapbox<.98\wd\linebox
      si-\box\snapbox
    \else
      no-\box\linebox
    \fi
  \fi}
\vbox {\lipsum{1}\par\eatlines}
\bye
```

si- Stoici autem, quod finem bonorum in una virtute ponunt, si
no-sunt illorum; Erat enim Polemonis. Ut optime, secundum naturam afl
no-tum esse possit. Negat esse eam, inquit, propter se expetendam. Bea
no-autem esse in maximarum rerum timore nemo potest. Idem iste, inqua
no-de voluptate quid sentit? Non est ista, inquam, Piso, magna dissensi

Párrafos con lastbox

14.8 Dimensiones y Glue

En TEX, el espacio blanco vertical y horizontal puede ajustarse a través de “estiramiento” o “encogimiento”. Un espacio blanco ajustable se llama “glue”. Este capítulo trata todos los conceptos técnicos relacionados

con las dimensiones y el glue, y explica cómo se calcula la mala calidad de estiramiento o encogimiento de una cantidad determinada.

1. `\dimen`: Este es un prefijo para un registro de dimensión. Un registro de dimensión es básicamente una variable que puede contener una medida. Por ejemplo, puedes tener para asignar 5 puntos a la dimensión 0.
2. `\dimendef`: Este comando se usa para definir una secuencia de control para ser un sinónimo de un registro de dimensión. Es útil cuando quieres darle un nombre más descriptivo a una dimensión.
3. `\newdimen`: Este comando se utiliza para crear una nueva dimensión. Por ejemplo, puedes hacer algo como `\newdimen\mydimen\mydimen=10pt`. Esto crea una nueva dimensión llamada `\mydimen` y le asigna un valor de 10 puntos.
4. `\skip` y `\newskip`: Estos se refieren a un tipo especial de espacio, el "glue", que puede estirarse o encogerse. `\newskip` se utiliza para crear un nuevo registro de este tipo de espacio.
5. `\hskip` y `\vskip`: Estos comandos insertan espacios horizontales y verticales, respectivamente. Por ejemplo, `\hskip` 1in insertará un espacio horizontal de 1 pulgada.
6. `\hfil`, `\hfill`, `\hss`: Estos son comandos para insertar espacios horizontales flexibles, que se estiran o encogen para llenar el espacio disponible. `\hfil` se estira o encoge a una tasa moderada, mientras que `\hfill` se estira o encoge más agresivamente. `\hss` puede encogerse en ambas direcciones.
7. `\vfil`, `\vfill`, `\vss`: Estos son equivalentes a `\hfil`, `\hfill` y `\hss` pero para espacios verticales.
8. `\kern`: Este comando agrega un espacio que no se estira ni encoge. Por ejemplo, puedes escribir `Hello\kern1inWorld` para obtener "Hello", seguido de un espacio de una pulgada, seguido de "World".
9. `\lastkern` y `\lastskip`: Estos comandos te permiten acceder al último espacio (ya sea un kern o un glue) que se insertó en la lista actual.
10. `\unkern` y `\unskip`: Estos comandos eliminan el último espacio (ya sea un kern o un glue) que se insertó en la lista actual.
11. `\advance`, `\multiply`, `\divide`: Estos son comandos aritméticos que te permiten modificar el valor de las dimensiones. Por ejemplo, `\advance\mydimen by 2pt` aumentará el valor de `\mydimen` en 2 puntos.

Definición de glue y dimensiones

Definición de dimensiones

Ejemplo de definición de dimensiones:

```
\newdimen\mydimen
\mydimen=1in % asigna 1 pulgada a \mydimen
\mydimen2in % asigna 2 pulgadas a \mydimen
```

estructura general del comando es

```
<dimen> -> <optional signs><unsigned dimen>
```

Definición de glue

Ejemplo de definición de glue:

```
\newskip\myskip
\myskip=10pt plus 1pt minus 1pt
```

14.9 Macros

Las macros en TeX son el mecanismo de abreviación para secuencias de comandos que se necesitan más de una vez, algo similar a las funciones o procedimientos en los lenguajes de programación convencionales. Estos comandos ya se han estudiado anteriormente, por lo que veremos solo aquellos comandos que no se profundizaron.

1. `\def`: Este comando inicia la definición de una macro.
2. `\gdef`: Este es un sinónimo para `\global\def`, que define una macro que es accesible globalmente en todo el documento, en lugar de solo en el ámbito en el que se define.
3. `\edef`: Este comando inicia la definición de una macro, pero expande el texto de reemplazo en el momento de la definición. Esto significa que cualquier macro dentro del texto de reemplazo también será expandida.
4. `\xdef`: Este es un sinónimo para `\global\edef`, que combina las propiedades de `\gdef` y `\edef`.
5. `\csname` y `\endcsname`: Estos comandos se utilizan para formar el nombre de una secuencia de control. Permiten crear nombres de secuencias de control que contienen caracteres no permitidos normalmente, o que son el resultado de la expansión de otras macros.
6. `\global`: Este comando hace que la siguiente definición, sentencia aritmética o asignación sea global, es decir, accesible en todo el documento.
7. `\outer`: Este prefijo indica que la macro que se está definiendo solo puede ser utilizada en el nivel "externo", es decir, no puede ser utilizada dentro de otra macro.
8. `\long`: Este prefijo indica que los argumentos de la macro que se está definiendo pueden contener tokens `\par`, lo que permite que los argumentos de la macro contengan párrafos.
9. `\let`: Este comando define una secuencia de control para que sea equivalente al siguiente token. Por ejemplo, podrías definir una nueva secuencia de control para ser equivalente a otra existente:

Construcción de secuencias de control

Los comandos \csname y \endcsname en TeX se utilizan para construir secuencias de control, que son esencialmente las "funciones" o "comandos" en TeX. Por ejemplo, el código

```
\csname hskip\endcsname 5pt
```

Esto es equivalente a \hskip5pt. La secuencia de control hskip se está creando usando \csname y \endcsname.

Durante este proceso de construcción, todas las macros y otras secuencias de control expandibles entre \csname y \endcsname se expanden como de costumbre, hasta que solo queden tokens de caracteres no expandibles.

El código:

```
\csname \ifhmode h\else v\fi skip\endcsname 5pt
```

Entra en modo horizontal (\ifhmode), se realiza un \hskip; si no, se realiza un \vskip.

Ejemplo de aplicación del kernel de LaTeX:

```
\def\newcounter#1{%
  \expandafter\@ifdefinable \csname c@#1\endcsname
    {\@definecounter{#1}}%
  \@ifnextchar[{\@newctr{#1}}{}}
```

14.10 Expansión

La expansión en TeX es bastante diferente de las llamadas a procedimientos en la mayoría de los lenguajes de programación.

A continuación se explican los comandos mencionados:

- \relax: Este comando no hace nada y se utiliza para indicar que no hay nada que expandir o ejecutar.
- \expandafter: Este comando toma los dos tokens siguientes y coloca la expansión del segundo token después del primer token. En otras palabras, \expandafter permite que el segundo token sea expandido antes del primero.
- \noexpand: Este comando se utiliza para indicar que el siguiente token no debe ser expandido.
- \edef: Este comando se utiliza para definir una macro, donde el texto de reemplazo se expande en el momento de la definición. En otras palabras, \edef toma la expansión de su argumento y la define como el texto de reemplazo de la macro.

- \aftergroup: Este comando se utiliza para guardar el siguiente token para su inserción después del grupo actual. En otras palabras, \aftergroup toma el siguiente token y lo inserta en el lugar correcto después del grupo actual.
- \afterassignment: Este comando se utiliza para guardar el siguiente token para su ejecución después de la siguiente asignación o definición de macro. En otras palabras, \afterassignment toma el siguiente token y lo ejecuta después de la siguiente asignación o definición de macro.
- \the: Este comando se utiliza para expandir el valor de varias cantidades en TEXen una cadena de tokens de caracteres. Por ejemplo, \the\count1 expandirá el valor del contador 1 en una cadena de tokens de caracteres.

14.11 Condicionales

Los condicionales son una herramienta esencial para crear macros potentes en TEX. TEXtiene una amplia gama de condicionales para consultar cosas como los códigos de categoría o los modos de procesamiento.

A continuación se describen los diferentes tipos de condicionales disponibles en TEX:

1. \if: Prueba la igualdad de los códigos de caracteres.
2. \ifcat: Prueba la igualdad de los códigos de categoría.
3. \ifx: Prueba la igualdad de la expansión de macros, o la igualdad de los códigos de caracteres y categoría.
4. \ifcase: Una declaración de casos enumerados.
5. \ifnum: Prueba las relaciones entre números.
6. \ifodd: Prueba si un número es impar.
7. \ifhmode: Prueba si el modo actual es (posiblemente restringido) modo horizontal.
8. \ifvmode: Prueba si el modo actual es (posiblemente interno) modo vertical.
9. \ifmmode: Prueba si el modo actual es (posiblemente de display) modo matemático.
10. \ifinner: Prueba si el modo actual es un modo interno.
11. \ifdim: Compara dos dimensiones.
12. \ifvoid: Prueba si un registro de caja está vacío.
13. \ifhbox: Prueba si un registro de caja contiene una caja horizontal.
14. \ifvbox: Prueba si un registro de caja contiene una caja vertical.
15. \ifeof: Prueba si se ha alcanzado el final de la secuencia de entrada o si un archivo no existe.

16 `\iftrue`: Una prueba que siempre es verdadera.

17 `\iffalse`: Una prueba que siempre es falsa.

18 `\fi`: Delimitador de cierre para todos los condicionales.

19 `\else`: Selecciona el texto falso de un condicional o el caso predeterminado de `\ifcase`.

20 `\or`: Separador para las entradas de un `\ifcase`.

21 `\newif`: Crea una nueva prueba.

14.12 Listas de Tokens

En TeX, las listas de tokens son la única estructura de datos que se maneja. Los tokens son los componentes más básicos de TeX: son como las "letras" del "alfabeto" de TeX. Cada control de secuencia, cada carácter y cada espacio en blanco es un token en TeX.

Existen registros de listas de tokens disponibles para el usuario, y TeX tiene algunas listas de tokens especiales, como las variables `\every`..., `\errhelp` y `\output`.

1 `\toks`: Este es el prefijo para un registro de lista de tokens. Por ejemplo, puedes usar `\toks0` para referirte al primer registro de lista de tokens.

2 `\toksdef`: Este comando define una secuencia de control para ser un sinónimo de un registro `\toks`. Por ejemplo, `\toksdef\mytoks=0` hará que `\mytoks` sea un sinónimo de `\toks0`.

3 `\newtoks`: Esta es una macro que asigna un registro de lista de tokens. Por ejemplo, `\newtoks\mytoks` asignará un nuevo registro de lista de tokens y lo llamará `\mytoks`.

Ejemplo 14.60

```
\pdfpagewidth=11cm \hsize=11cm \hoffset=-1in \pdfpageheight=3cm \vsize=3cm \voffset=-1in
\newtoks\mytoks
\mytoks={Este es un ejemplo de una lista de tokens.}
\the\mytoks
\bye
```

Este es un ejemplo de una lista de tokens.

Este ejemplo crea un nuevo registro de lista de tokens llamado `\mytoks`, asigna una lista de tokens a `\mytoks` y luego muestra el contenido de `\mytoks`.

Lista de tokens

TeX tiene 256 registros de lista de tokens, llamados `\toks nnn`, que pueden ser asignados usando la macro `\newtoks`, o explícitamente con `\toksdef`.

Veamos un poco más en detalle estos conceptos con algunos ejemplos.

Token: Un token es la unidad básica de información en TeX. Puede ser un carácter o una secuencia de control. Por ejemplo, en el texto `\hskip 5cm`, `\hskip` es un token de secuencia de control, y `5`, `c` y `m` son tokens de caracteres.

Lista de Tokens: Una lista de tokens es simplemente una secuencia de tokens. Por ejemplo, en el texto `\hskip 5cm`, la lista de tokens es `[\hskip, 5, c, m]`.

Registro de tokens

TeX tiene 256 registros donde puedes almacenar listas de tokens. Puedes pensar en estos registros como variables en las que puedes almacenar listas de tokens.

`\newtoks`: Esta es una macro que te permite asignar un nuevo registro de lista de tokens. Por ejemplo, el código `\newtoks\myTokens` asigna un nuevo registro de lista de tokens llamado `\myTokens`.

`\toksdef`: Este es un comando que te permite asignar explícitamente un registro de lista de tokens a una secuencia de control. Por ejemplo, `\toksdef\myTokens=0` asigna el registro de lista de tokens 0 a la secuencia de control `\myTokens`.

Desempaquetar una lista de tokens

El comando `\the` se usa para "desempaquetar" una lista de tokens, lo que significa obtener los tokens individuales de la lista. Entonces, si `\toks0` contiene la lista de tokens `{a, espacio, b, espacio, c, d}`, entonces `\the\toks0` producirá la secuencia de tokens `a, espacio, b, espacio, c, d`.

token parameter

Es una lista de tokens en TEXque se insertan automáticamente en ciertos puntos. Estos ⟨token parameters⟩ son los siguientes:

1. `\output`: Esta lista de tokens se inserta cada vez que TEXdecide que tiene suficiente material para una página o cuando el usuario fuerza la activación mediante una penalización `<=-10000` en modo vertical.

2. `\everypar`: Se inserta cuando TEXcambia del modo vertical externo o interno a un modo horizontal sin restricciones.

3. `\everymath`: Se inserta después de un solo carácter de cambio de fórmula que inicia una fórmula.

4 `\everydisplay`: Se inserta después de un doble carácter de cambio de fórmula que inicia una fórmula de visualización.

5 `\everyhbox`: Se inserta cuando comienza una `\hbox`.

6 `\everyvbox`: Se inserta cuando comienza una caja vertical.

7 `\everyjob`: Se inserta cuando comienza un trabajo.

8 `\everycr`: Se inserta en alineaciones después de `\cr` o un `\crcr` no redundante.

9 `\errhelp`: Contiene tokens para complementar un `\errmessage`.

14.13 Distancias de línea base

Estas son las definiciones de algunos conceptos clave relacionados con la alineación de texto en TEX.

1 `\baselineskip`: La distancia ideal entre las líneas de texto en una lista vertical.

2 `\lineskiplimit`: La distancia mínima que debe mantenerse entre las líneas de texto en una lista vertical.

3 `\lineskip`: La cantidad de "pegamento" añadido si la distancia entre las líneas es menor que `\lineskiplimit`.

4 `\prevdepth`: La profundidad del último cuadro agregado a una lista vertical, tal como lo percibe TEX.

5 `\nointerlineskip`: Una macro para prevenir la inserción de pegamento interlineal una sola vez.

6 `\offinterlineskip`: Una macro para prevenir la inserción de pegamento interlineal de forma global a partir de ahora.

7 `\openup`: Aumenta `\baselineskip`, `\lineskip` y `\lineskiplimit` en la cantidad especificada.

Observaciones adicionales

En general, para documentos más largos de una página es deseable tener la misma distancia de línea base en todo el documento. Sin embargo, para documentos de una sola página, puede agregar estirabilidad a la distancia de línea base, por ejemplo, si el texto tiene que estar alineado en la parte inferior.

Se puede aumentar la distancia entre solo un par de líneas con `\vadjust`. El argumento de este comando es material vertical que se insertará en la lista vertical justo después de la línea donde se dio este comando. La segunda línea de este párrafo, por ejemplo, contiene el comando `\vadjust{\kern2pt}`.

La cantidad de espacio entre líneas no se puede cambiar en medio de un párrafo, ya que el valor para `\baselineskip` que se usa es el que se utiliza cuando el párrafo finalmente se rompe y se agrega a la lista vertical principal. Lo mismo se aplica para `\lineskip` y `\lineskiplimit`.

La macro de TEX simple \openup aumenta \baselineskip, \lineskip y \lineskiplimit en la cantidad del argumento de la macro. En efecto, esto aumenta las distancias entre líneas en esta cantidad independientemente de si están reguladas por \baselineskip o \lineskip.

14.14 Inicio de párrafo

En el inicio de un nuevo párrafo, TEXinserta un salto vertical como separación del párrafo anterior y un salto horizontal como sangría para el párrafo actual. En este sección se estudiará la secuencia exacta de acciones y discute cómo se pueden alterar las decisiones de TEX.

1. \indent: Cambia al modo horizontal e inserta una caja de ancho \parindent.
2. \noindent: Cambia al modo horizontal con una lista horizontal vacía.
3. \parskip: Cantidad de pegamento agregada a la lista vertical circundante cuando comienza un párrafo. Por defecto en TEXPlano: 0pt más 1pt.
4. \parindent: Tamaño de la caja de sangría insertada delante de un párrafo. Por defecto en TEXPlano: 20pt.
5. \everypar: Lista de tokens insertada delante del texto de un párrafo.
6. \leavevmode: Macro para cambiar al modo horizontal si es necesario.

Inicio de párrafo

TEX genera un nuevo párrafo cuando cambia de modo vertical a modo horizontal sin restricciones. Este cambio puede ser causado por los comandos \indent y \noindent. Por ejemplo, si tienes el siguiente código:

```
{\bf Hello, world!}
\vskip 2cm
\noindent Hello, world!
```

Aquí, la cadena {\bf Hello, world!} está en modo horizontal. Después de un salto vertical (\vski␣p3pt), TEX vuelve al modo vertical. Entonces, \noindent hace que TEX entre en el modo horizontal y comienza un nuevo párrafo sin indentación.

Además de \indent y \noindent, hay otras "órdenes horizontales" que pueden desencadenar el comienzo de un nuevo párrafo. Por ejemplo:

```
\vskip3pt
It's~\dots
```

Aquí, la "I" es una orden horizontal y provoca que TEX inicie un nuevo párrafo. Lo mismo ocurre con un único $ y \hskip:

```
$x$ is supposed~\dots
```

x hace que TEX inicie un nuevo párrafo.

leavevmode

El comando \leavevmode se define en LaTeX y plainTeX y asegura que se finalice el modo vertical y se ingrese en el modo horizontal. En el modo vertical, TeX apila cajas horizontales verticalmente, mientras que en el modo horizontal, se toman como parte de la línea de texto.

Por ejemplo, \leavevmode se define para asegurarse de que se ingrese en el modo horizontal si se usa al comienzo de un párrafo. Si solo lo usa \hbox{...}, se apila encima del siguiente párrafo en su lugar.

Ejemplo 14.61

```
Text\par\hbox{Hello} World Resultado:
```

Text
Hello
World Resultado:

Puede ver que en el primer ejemplo, \hbox{...} se apila con los dos párrafos verticalmente (pero sin sangría de párrafo) porque se procesa en modo vertical. En el segundo caso, primero se ingresa en el modo horizontal y luego se procesa como parte del segundo párrafo.

\hbox es un comando de TeX que crea una caja horizontal. El contenido de la caja está contenido dentro de llaves y se coloca en una línea horizontal.

14.15 Fin de párrafo

En esta sección se estudia los mecanismos para finalizar un párrafo.

1. \par: Finaliza un párrafo y pasa al modo vertical.
2. \endgraf: Sinónimo de \par. \let\endgraf=\par
3. \parfillskip: Pegamento que se coloca entre el último elemento del párrafo y el final de la línea. Por defecto en TEXPlano: 0pt más 1fil.

Después del comando \par, TeX entra en modo vertical y ejerce el constructor de página. Si el \par fue insertado porque un comando vertical ocurrió en modo horizontal, el comando vertical es entonces examinado nuevamente. El \par no inserta ningún pegamento vertical ni penalizaciones por sí mismo. Un comando \par también borra los parámetros de forma del párrafo.

Es importante distinguir entre el token \par y el comando primitivo \par que es el significado inicial de ese token. El token \par se inserta cuando el procesador de entrada ve una línea vacía, o cuando el procesador de ejecución encuentra un comando vertical en modo horizontal; el comando \par es lo que realmente cierra un párrafo.

Después del último elemento del párrafo, TeX inserta implícitamente el equivalente de `\unskip \penalty10000 \hskip\parfillskip`. El `\unskip` sirve para eliminar cualquier pegamento espurio al final del párrafo, como el espacio generado por el fin de línea si el `\par` fue insertado por el procesador de entrada.

El `\parfillskip` es en TEX plano de primer orden infinito (0pt más 1fil), por lo que terminar un párrafo con `\hfil$\bullet$\par` dará un punto a mitad de camino entre la última palabra y el final de la línea; con `\hfill$\bullet$\par` estará alineado a la derecha.

Si un párrafo se establece en un grupo, puede ser necesario asegurarse de que el `\par` que termina el párrafo ocurre dentro del grupo. Los parámetros que influyen en la composición tipográfica del párrafo, como el `\leftskip` y el `\baselineskip`, sólo se miran cuando se termina el párrafo.

La secuencia `\hfill\break` es una forma de forzar una 'nueva línea' dentro de un párrafo. Si terminas un párrafo con esto, probablemente obtendrás un error de Caja Inferior Llena.

El comando `\par` no inserta ningún pegamento en la lista vertical, por lo que en la secuencia

14.16 Forma de párrafo

Este capítulo trata sobre los parámetros y comandos que influyen en la forma de un párrafo.

1. `\parindent`: Es la anchura de la caja de sangría que se agrega al principio de un párrafo. El valor predeterminado en TEXplano es 20pt.
2. `\hsize`: Es la anchura de línea utilizada para componer un párrafo. El valor predeterminado en TEXplano es 6.5 pulgadas.
3. `\leftskip`: Es el espacio que se coloca a la izquierda de todas las líneas de un párrafo.
4. `\rightskip`: Es el espacio que se coloca a la derecha de todas las líneas de un párrafo.
5. `\hangindent`: Si es positivo, indica la sangría desde el margen izquierdo; si es negativo, es la negación de la sangría desde el margen derecho.
6. `\hangafter`: Si es positivo, indica el número de líneas antes de que comience la sangría; si es negativo, su valor absoluto indica el número de líneas sangradas comenzando con la primera línea del párrafo. Valor predeterminado: 1.
7. `\parshape`: Es el comando para formas generales de párrafo.

Ancho de línea de texto

Imagínate que TEX es un artesano que está creando un mosaico de palabras. Cada párrafo es una fila en este mosaico, y las palabras son las piezas que se están colocando.

Cuando TEX termina de leer un párrafo, ha recogido todas las palabras (formando una "lista horizontal") y está listo para comenzar a colocarlas en su lugar.

Primero, coloca un "espacio de sangría" (una caja de sangría) al comienzo. Luego, comienza a colocar las palabras. Cada línea que forma tiene exactamente la longitud de \hsize, que es como la longitud de la fila en el mosaico.

Para hacer que las palabras encajen perfectamente en esta longitud, TEX puede añadir un poco de "pegamento" (espacio adicional) a ambos lados de la línea. Los comandos \leftskip y \rightskip te permiten controlar cuánto de este pegamento se añade en los lados izquierdo y derecho de la línea, respectivamente.

Por otro lado, los comandos \parshape y \hangindent te permiten modificar la longitud de \hsize y desplazar las líneas, respectivamente, proporcionando aún más control sobre cómo se colocan las palabras en cada línea.

Ejemplo 14.62

```
\pdfpagewidth=11cm \hsize=11cm \hoffset=-1in \pdfpageheight=3cm \vsize=3cm \voffset=-1in
\leftskip=0.5in
\rightskip=0.5in
No importa cuánto texto escribas aquí, será configurado en un bloque de texto con una longitud de 1 cm
↪ (hsize), con una sangría de 0.5 pulgadas a la izquierda y 0.5 pulgadas a la derecha. Esto es útil
↪ para configurar el diseño de tus párrafos.
\bye
```

No importa cunto texto escribas aqu, ser configurado en un bloque de texto con una longitud de 1 cm (hsize), con una sangra de 0.5 pulgadas a la izquierda y 0.5 pulgadas a la derecha. Esto es til para configurar el diseo de tus prrafos.

Ejemplo 14.63

```
\pdfpagewidth=11cm \hsize=11cm \hoffset=-1in \pdfpageheight=3cm \vsize=3cm \voffset=-1in
\parshape=3
0.5in 4.5in
1in 3.5in
2in 1.5in
Este es un párrafo que tendrá una forma particular. La primera línea comenzará 0.5 pulgadas a la
↪ derecha y tendrá una longitud de 4.5 pulgadas. La segunda línea comenzará 1 pulgada a la derecha y
↪ tendrá una longitud de 3.5 pulgadas. La tercera línea (y todas las siguientes, si las hay)
↪ comenzará 2 pulgadas a la derecha y tendrá una longitud de 1.5 pulgadas.
\bye
```

Este es un prrafo que tendr una forma particular. La pr
comenzar 0.5 pulgadas a la derecha y tendr una lon
tud de 4.5 pulgadas. La
segunda lnea comenzar 1
pulgada a la derecha y tendr▮
una longitud de 3.5 pul-
gadas. La tercera lnea (y

todas las siguientes, si las
hay) comenzar 2 pulgadas▮
a la derecha y tendr una
longitud de 1.5 pulgadas.

Ejemplo 14.64

```
\pdfpagewidth=11cm \hsize=11cm \hoffset=-1in \pdfpageheight=2cm \vsize=2cm \voffset=-1in
\hangindent=0.5in
Esto es un párrafo con sangría. La primera línea comenzará en el margen establecido, pero todas las
↪  demás líneas del párrafo estarán 0.5 pulgadas a la derecha del margen.
\bye
```

Esto es un prrafo con sangra. La primera lnea comenzar en el margen establecido, pero todas las dems lneas del prrafo estarn 0.5 pulgadas a la derecha del margen.

Shape parameters

General paragraph shape

El formato general de \parshape es:

$$n\, i_1\, l_1 \ldots i_n l_n$$

```
\parshape=n i1 l1 i2 l2 ... in ln
```

donde n es el número de especificaciones de línea que sigue, y cada par i y l representa la sangría (i) y la longitud de la línea (l) para cada una de las primeras n líneas.

Por ejemplo, si quieres un párrafo en el que la primera línea tenga una sangría de 1cm y una longitud de 4cm, y la segunda línea tenga una sangría de 2cm y una longitud de 3cm, puedes usar:

```
\parshape=2 1cm 4cm 2cm 3cm
Este es un párrafo de forma especial ...
```

Si el párrafo tiene más de n líneas, se repetirá la especificación de la última línea. Si el párrafo tiene menos de n líneas, se ignorarán las especificaciones sobrantes.

El valor predeterminado es \parshape=0, que significa que no se aplican reglas especiales de forma de párrafo.

Un comando \parshape anula un \hangindent si ambos han sido especificados. Además, \parshape no impide que se apliquen los comandos regulares \parindent, \leftskip y \rightskip.

Al igual que \hangindent, \hangafter y \looseness, el parámetro \parshape se borra después de un comando \par. Dado que cada línea vacía genera un token \par, no debes dejar una línea vacía entre una declaración de forma de párrafo (o sangría colgante) y el párrafo siguiente.

Por último, \parshape es un “entero interno”. Eso significa que si consultas su valor (\the\parshape), obtendrás el número de líneas n con el que fue establecido.

Ejemplo 14.65

```
\documentclass{article}
\usepackage[margin=5mm,paperwidth=5.4cm,paperheight=8.5cm]{geometry}
\begin{document}
\smallskip\font\sixrm=cmr6 \sixrm \baselineskip=7pt
\fontdimen3\font = 1.8pt \fontdimen4\font = 0.9pt
\noindent \hfuzz 0.1pt
\parshape 30 0pt 120pt 1pt 118pt 2pt 116pt 4pt 112pt 6pt 108pt 9pt 102pt 12pt 96pt 15pt 90pt 19pt 84pt
↪ 23pt 77pt 27pt 68pt 30.5pt 60pt 35pt 52pt 39pt 45pt 43pt 36pt 48pt 27pt 51.5pt 21pt 53pt 16.75pt
↪ 53pt 16.75pt 53pt 16.75pt 53pt 16.75pt 53pt 16.75pt 53pt 16.75pt 53pt 16.75pt 53pt 16.75pt 53pt
↪ 14.6pt 48pt 24pt 45pt 30.67pt 36.5pt 51pt 23pt 76.3pt The wines of France and California may be the
↪ best known, but they are not the only fine wines. Spanish wines are often underestimated, and quite
↪ old ones may be available at reasonable prices. For Spanish wines the vintage is not so critical,
↪ but the climate of the Bordeaux region varies greatly from year to year. Some vintages are not as
↪ good as others, so these years ought to be s\kern -.1pt p\kern -.1pt e\kern -.1pt c\hfil ially
↪ n\kern .1pt o\kern .1pt t\kern .1pt e\kern .1pt d\hfil: 1962, 1964, 1966. 1958, 1959, 1960, 1961,
↪ 1964, 1966 are also good California vintages. Good luck finding them!
\end{document}
```

The wines of France and California
may be the best known, but they
are not the only fine wines. Span-
ish wines are often underestimated,
and quite old ones may be avail-
able at reasonable prices. For
Spanish wines the vintage is
not so critical, but the cli-
mate of the Bordeaux re-
gion varies greatly from
year to year. Some
vintages are not as
good as others,
so these years
ought to be
specially
noted:
1962,
1964,
1966.
1958,
1959,
1960,
1961,
1964,
1966
are also
good Cal-
ifornia vintages.
Good luck finding them!

14.17 Romper línea

Este extracto está discutiendo la forma en que TeX gestiona la división de texto en líneas y páginas, utilizando un concepto llamado "badness" (maldad) para decidir dónde son los mejores lugares para dividir un párrafo en líneas o una página en secciones.

Hay una serie de penalizaciones que contribuyen a la "badness", y estos parámetros permiten ajustar cómo TeX toma estas decisiones. Aquí hay una descripción de cada uno de ellos, con un ejemplo de cómo se usarían en un documento de TeX:

1. `\penalty`: Especifica la deseabilidad de no hacer una división en este punto. Por ejemplo, puedes usar `\penalty10000` para evitar una división en un lugar específico.
2. `\linepenalty`: Penalización asociada a cada salto de línea. El valor por defecto en TeX plano es 10.

3 \hyphenpenalty: Penalización asociada a un salto en un elemento discrecional (por lo general, un guión). El valor por defecto en TeX plano es 50.

4 \exhyphenpenalty: Penalización por hacer un salto de línea en un elemento discrecional, en el caso especial donde el texto pre-break está vacío. El valor por defecto en TeX plano es 50.

5 \adjdemerits: Penalización por líneas visualmente incompatibles adyacentes. El valor por defecto en TeX plano es 10000.

6 \doublehyphendemerits: Penalización por líneas consecutivas que terminan con un guión. El valor por defecto en TeX plano es 10000.

7 \finalhyphendemerits: Penalización añadida cuando la penúltima línea de un párrafo termina con un guión. El valor por defecto en TeX plano es 5000.

8 \allowbreak: Macro para crear un punto de quiebre insertando un \penalty0.

9 \pretolerance: Valor de tolerancia para un párrafo sin guiones. El valor por defecto en TeX plano es 100.

10 \tolerance: Valor de tolerancia para líneas en un párrafo con guiones. El valor por defecto en TeX plano es 200.

11 \emergencystretch: (sólo TEX3) Se asume una estirabilidad extra en las líneas de un párrafo.

12 \looseness: Número de líneas por las cuales este párrafo tiene que ser hecho más largo de lo que sería idealmente.

13 \prevgraf: El número de líneas en el último párrafo añadido a la lista vertical.

14 \discretionary: Especifica la forma en que una secuencia de caracteres se divide en un salto de línea.

15 \-: Guion discrecional; esto es equivalente a \discretionary{-}{}{}.

16 \hyphenchar: Número del carácter de guión de una fuente.

17 \defaulthyphenchar: Valor de \hyphenchar cuando se carga una fuente. El valor por defecto en TeX plano es \-.

18 \uchyph: Positivo para permitir la separación de palabras que comienzan con una letra mayúscula. El valor por defecto en TeX plano es 1.

19 \lefthyphenmin: (sólo TEX3) Número mínimo de caracteres antes de una separación. El valor por defecto en TeX plano es 2.

20 \righthyphenmin: (sólo TEX3) Número mínimo de caracteres después de una separación. El valor por defecto en TeX plano es 3.

21 `\patterns`: Define una lista de patrones de separación para el valor actual de `\language`; sólo permitido en IniTEX.

22 `\hyphenation`: Define excepciones de separación para el valor actual de `\language`.

23 `\language`: Selecciona un conjunto de patrones de separación y excepciones.

24 `\setlanguage`: Reinicia el idioma actual.

Ejemplo:

```
\linepenalty=200
```

Esta línea establece el valor de la penalidad asociada con cada salto de línea en 200. Esto significa que TEXserá más propenso a evitar un salto de línea en este punto si la penalidad asociada con él es mayor que 200.

Ejemplo:

```
\hyphenpenalty=150
```

Esta línea establece el valor de la penalidad asociada con una interrupción de hiato en 150. Esto significa que TEXserá más propenso a evitar una interrupción de hiato en este punto si la penalidad asociada con él es mayor que 150.

Ejemplo:

```
\pretolerance=80
```

Esta línea establece el valor de la tolerancia para un párrafo sin hiperfijación en 80. Esto significa que TEXtratará de hacer que cada línea en el párrafo sea al menos tan larga como 80% de la longitud de la línea ideal.

Ejemplo:

```
\discretionary{-}{}{}
```

Esta línea especifica el modo en que una secuencia de caracteres se divide en un salto de línea. El ejemplo especifica que, en caso de necesidad de un salto de línea en este punto, se insertará un guión hacia la derecha (-) y los caracteres restantes irán a la línea siguiente.

Ejemplo:

```
\language=english
```

Esta línea elige un conjunto de patrones y excepciones de hiperfijación para el idioma inglés. Esto significa que TEXutilizará los patrones y excepciones de hiperfijación definidos para el idioma inglés para determinar cómo dividir palabras en líneas.

14.18 Espaciado

La distancia entre palabras típica en TEX se especifica en la información de la fuente, pero el usuario puede anular esta configuración. En esta sección se explica las reglas según las cuales TEX calcula la distancia entre palabras.

1. `\␣` espacio de control: inserta la misma cantidad de espacio que insertaría un token de espacio si `\spacefactor = 1000`.
2. `\spaceskip`: pegamento o glue interpalabra si es distinto de cero.
3. `\xspaceskip`: pegamento o glue interpalabra si es distinto de cero y `\spacefactor >= 2000`.
4. `\spacefactor`: 1000 veces la relación por la que se multiplica (se divide) el componente de estiramiento (contracción) del pegamento interpalabra.
5. `\sfcode`: Valor de `\spacefactor` asociado con un carácter.
6. `\frenchspacing`: macro para desactivar el espacio extra después de la puntuación.
7. `\nonfrenchspacing`: macro para activar el espacio extra después de la puntuación.

14.19 Alineación

1. `\halign` y `\valign`: Estos comandos se usan para la alineación horizontal y vertical respectivamente. Ambos toman un conjunto de "plantillas" para cada columna, y luego los valores de esas columnas se llenan con la información proporcionada. Por ejemplo:
2. `\omit`: Este comando se utiliza para omitir la plantilla de una entrada de alineación. Por ejemplo: En este caso, "Mundo" no se ajustará a la plantilla porque está precedido por `\omit`.
3. `\span`: Este comando se utiliza para unir dos entradas de alineación adyacentes.
4. `\multispan`: Esta macro se usa para unir un número de entradas de alineación adyacentes.
5. `\tabskip`: Este comando establece la cantidad de espacio entre columnas (filas) de un `\halign` (`\valign`).
6. `\noalign`: Este comando se usa para especificar material vertical (horizontal) que se colocará entre las filas (columnas) de un `\halign` (`\valign`).
7. `\cr` y `\crcr`: Estos comandos se usan para terminar una línea de alineación. `\crcr` solo tiene efecto si la línea no ha sido ya terminada por `\cr`.
8. `\everycr`: Esta es una lista de tokens que se inserta después de cada `\cr` o `\crcr` no redundante.
9. `\centering`: Este registro de pegamento en TEX plano se utiliza para centrar `\eqalign` y `\eqalignno`.

10 `\hideskip`: Este registro de pegamento en TEX plano se utiliza para hacer invisibles las entradas de alineación.

11 `\hidewidth`: Esta macro se utiliza para hacer invisible la entrada anterior o posterior.

Alineación horizontal y vertical

Los comandos `\halign` y `\valign` de TeX permiten controlar la alineación de elementos, principalmente en la creación de tablas.

`\halign`: Esta es una orden que permite alinear horizontalmente las columnas de una tabla. Al usar `\halign`, se crea una lista de cajas horizontales (es decir, filas) que luego se colocan en una lista vertical. Para entender esto, piensa en una tabla común: se compone de filas que están dispuestas verticalmente una tras otra. Así es como funciona `\halign`.

Un ejemplo sencillo de cómo funciona `\halign` sería este:

Ejemplo 14.66

```
\pdfpagewidth=11cm \hsize=11cm \hoffset=-1in \pdfpageheight=2cm \vsize=2cm \voffset=-1in
\halign{
  #&#\cr
  1&2\cr
  3&4\cr
}
\bye
```

12
34

Este fragmento de código genera una tabla de 2x2. Los números 1 y 2 se convierten en la primera fila de la tabla, mientras que los números 3 y 4 se convierten en la segunda fila. Las columnas se alinean horizontalmente gracias al uso de `\halign`.

En este caso, el carácter # se utiliza como marcador de posición para las entradas de la tabla, & se utiliza para separar las entradas de la tabla, y `\cr` se utiliza para marcar el final de una fila.

`\valign`: El comando `\valign` funciona de manera similar a `\halign`, pero en lugar de alinear las columnas horizontalmente, `\valign` alinea las filas verticalmente. Entonces, `\valign` crea una lista de cajas verticales (columnas) que luego se colocan en una lista horizontal.

Ejemplo 14.67

```
\pdfpagewidth=11cm \hsize=11cm \hoffset=-1in \pdfpageheight=2cm \vsize=2cm \voffset=-1in
\valign{
  #     &#       &#\cr
```

```
  1     &2      &3\cr
}
\bye
```

1
2
3

Material entre líneas

Hay ocasiones en las que puede que necesites añadir material adicional en la alineación, pero que no debería tratarse como una entrada o serie de entradas. Esto se puede lograr usando el comando `\noalign`.

Por ejemplo, para dibujar una línea horizontal entre dos filas en un `\halign`, puedes usar `\noalign{\hrule}`. Para desalentar un salto de página o de línea entre dos filas o columnas en un `\halign` o `\valign`, puedes usar `\noalign{\penalty100}`.

Tamaño de la alineación

Si necesitas darle a la alineación un tamaño predeterminado, puedes usar una especificación de caja. Por ejemplo, puedes usar `\halign to \hsize{ ... }` para hacer que la alineación tenga el mismo tamaño que `\hsize`. Es importante tener en cuenta que cualquier pegamento (es decir, espacio) contenido en las entradas de la alineación no tiene ningún papel en este ajuste. En lugar de eso, cualquier estiramiento o encogimiento requerido se toma del pegamento `\tabskip`.

El preambulo

Imagina que deseas crear una tabla simple con dos columnas, donde la primera columna está alineada a la izquierda y la segunda columna está centrada. Podrías hacer algo así:

Ejemplo 14.68

```
\pdfpagewidth=11cm \hsize=11cm \hoffset=-1in \pdfpageheight=2cm \vsize=2cm \voffset=-1in
\halign{#\hfil & \hfil#\hfil\cr
  Columna1 & Columna2 \cr
  1 & 2 \cr
  3 & 4 \cr
}
\bye
```

Columna1	Columna2
1	2
3	4

sto creará una tabla con dos columnas y tres filas. La primera fila contiene los encabezados de las columnas, "Columna1" y "Columna2". Las siguientes filas contienen los números 1 a 4 en las respectivas celdas.

Desglosemos el preámbulo aquí: `#\hfil & \hfil#\hfil\cr`

1. `#\hfil`: Esta es la plantilla para la primera columna. El # representa el contenido de la columna, y `\hfil` es un comando TeX que inserta un espacio horizontal flexible, empujando el contenido hacia la izquierda (alineado a la izquierda).
2. `&`: Este es el carácter de tabulación, que separa las columnas.
3. `\hfil#\hfil`: Esta es la plantilla para la segunda columna. Los espacios horizontales flexibles antes y después del # centran el contenido de la columna.

Ahora hablemos sobre el `\tabskip`. Imagina que quieres agregar algo de espacio adicional entre tus columnas. Podrías hacerlo con `\tabskip` así:

Ejemplo 14.69

```
\pdfpagewidth=11cm \hsize=11cm \hoffset=-1in \pdfpageheight=2cm \vsize=2cm \voffset=-1in
\tabskip=1em
\halign{#\hfil & \hfil#\hfil\cr
  Columna1 & Columna2 \cr
  1 & 2 \cr
  3 & 4 \cr
}
\bye
```

Columna1	Columna2
1	2
3	4

Este código añade 1em de espacio entre las columnas. em es una unidad de medida en TeX que es relativa al tamaño de fuente actual.

También se mencionó la alineación infinita, es decir, tener un número desconocido de columnas. Podrías tener un caso en el que tengas un número variable de columnas, en función de algunas condiciones. Aquí está cómo podrías hacerlo:

Ejemplo 14.70

```
\pdfpagewidth=11cm \hsize=11cm \hoffset=-1in \pdfpageheight=2cm \vsize=2cm \voffset=-1in
\halign{& \hfil#\hfil\cr
  Columna1 & Columna2 & Columna3 \cr
  1 & 2 & 3 \cr
  4 & 5 & 6 \cr
}
\bye
```

Columna1	Columna2	Columna3
1	2	3
4	5	6

En este ejemplo, la parte `& \hfil#\hfil` de la plantilla es repetible, lo que significa que puede repetirse para cualquier número de columnas.

14.20 Forma de página

1. `\topskip`: Este es la distancia entre la parte superior de la página y la primera línea de texto. Supongamos que desea aumentar esta distancia a 20pt. Lo haría de la siguiente manera:

```
\setlength{\topskip}{20pt}
```

   Luego, la primera línea de texto en cada página estaría a 20 puntos del borde superior.

2. `\hoffset` y `\voffset`: Estos comandos ajustan la posición horizontal y vertical de la página, respectivamente. Supongamos que quiere desplazar la página 1 cm a la derecha y 2 cm hacia abajo. Podría hacer esto así:

```
\setlength{\hoffset}{1cm}
\setlength{\voffset}{2cm}
```

3. `\vsize`: Este comando controla la altura de la caja de la página. Si desea reducir la altura de la caja de la página a 20 cm (quizás porque está imprimiendo en un tipo de papel más pequeño), haría esto:

```
\setlength{\vsize}{20cm}
```

4. `\maxdepth`: Este es la profundidad máxima de la caja de la página. Supongamos que tiene un elemento que se extiende debajo de la línea base del texto y no quiere que exceda de 5pt. Podría establecer el maxdepth de la siguiente manera:

```
\setlength{\maxdepth}{5pt}
```

5. `\splitmaxdepth`: Si estás usando el comando `\vsplit` para dividir cajas verticales, puedes controlar la profundidad máxima de la caja resultante con `\splitmaxdepth`. Si no quieres que ninguna caja dividida tenga una profundidad de más de 10pt, podrías hacer esto:

```
\setlength{\splitmaxdepth}{10pt}
```

14.21 Romper página

En esta sección trataremos sobre la construcción de páginas. La parte de TEXque decide dónde romper la lista vertical principal en páginas. El constructor de páginas opera antes de la rutina de salida y entrega su resultado en `\box`255 a la rutina de salida.

1. `\vsplit`: Esta instrucción divide la parte superior de una caja. Es similar a la creación de una nueva página.

2. `\splittopskip`: Esta longitud determina la distancia mínima entre la parte superior de lo que queda después de una operación `\vsplit` y el primer elemento en esa caja. Es 10pt por defecto en Plain TeX.

3. `\pagegoal`: Este registro determina la altura objetivo de la página. Comienza en `\vsize` y se reduce por las alturas de los elementos de inserción.

4. `\pagetotal`, `\pagedepth`, `\pagestretch`, `\pagefilstretch`, `\pagefillstretch`, `\pagefilllstretch`, `\pageshrink`: Estos registros almacenan distintas medidas acumulativas de la página actual en construcción. Por ejemplo, `\pagetotal` es la altura natural acumulada de la página actual, mientras que `\pagestretch` es el estiramiento de orden cero acumulado de la página actual.

5. `\outputpenalty`: Tiene el valor de la penalidad en el punto de ruptura de página actual. Si la ruptura no fue por una penalidad, su valor es 10000.

6. `\interlinepenalty`, `\clubpenalty`, `\widowpenalty`, `\displaywidowpenalty`, `\brokenpenalty`: Estos registros definen las penalidades para distintos tipos de rupturas de página. Por ejemplo, `\interlinepenalty` define una penalidad para romper una página entre líneas de un párrafo.

7. `\penalty`: Coloca una penalidad en la lista actual. Un ejemplo de su uso puede ser:

   ```
   Hello\penalty100 World
   ```

8. `\lastpenalty`: Si el último elemento de la lista era una penalidad, este es su valor.

9. `\unpenalty`: Este comando elimina el último elemento de la lista actual si este era una penalidad.

Los comandos relacionados con las penalidades se utilizan para ajustar la calidad de los puntos de ruptura de página o línea seleccionados por el algoritmo de TeX. Al aumentar una penalidad, estás desalentando a TeX a realizar una ruptura en ese punto. Por otro lado, al disminuir una penalidad, estás alentando a TeX a considerar esa posición como un buen punto de ruptura.

Página actual y contribuciones

La lista vertical principal de TeX se divide en dos partes: la "página actual" y la lista de "contribuciones recientes". Cualquier material que se añade a la lista vertical principal se añade a las contribuciones recientes; el acto de mover las contribuciones recientes a la página actual se conoce como "ejercer el manejador de páginas".

Por ejemplo, si añades una sección de texto a tu documento TeX, esa sección de texto se añade inicialmente a las contribuciones recientes. Luego, el manejador de páginas decide cuándo mover esa sección a la página actual.

Activando el generador de páginas

El manejador de páginas se activa en las siguientes circunstancias:

- Alrededor de los párrafos: después de que los tokens `\everypar` han sido insertados, y después de que el párrafo ha sido añadido a la lista vertical.
- Alrededor de fórmulas de visualización: después de que los tokens `\everydisplay` han sido insertados, y después de que la visualización ha sido añadida a la lista.
- Después de comandos `\par`, cajas, inserciones y penalizaciones explícitas en modo vertical.
- Después de que una rutina de salida ha terminado.

Contabilidad de la longitud de la página

La altura y la profundidad de la caja de la página que llega a la rutina de salida están determinadas por `\vsize`, `\topskip`, y `\maxdepth`.

Después de la primera caja, regla o inserción en la página actual, el `\vsize` se registra en `\pagegoal`, y su valor no se consulta hasta que `\output` ha estado activo. Cambiar `\pagegoal` sí tiene un efecto en la página actual. Cuando la página está vacía, el objetivo de la página es `\maxdimen`, y `\pagetotal` es cero.

Las dimensiones acumuladas y el estiramiento están disponibles en los parámetros `\pagetotal`, `\pagedepth`, `\pagestretch`, `\pagefilstretch`, `\pagefillstretch`, `\pageshrink`, y `\pagefilllstretch`. Se actualizan cada vez que se añade pegamento a la página. El parámetro de profundidad se convierte en cero si el último elemento fue kern o pegamento.

Puntos de roptura

Posibles puntos de roptura

Los saltos de página pueden ocurrir en los mismos tipos de lugares donde pueden ocurrir los saltos de línea:

1. En el pegamento que es precedido por un elemento no descartable: El pegamento en TeX es el espacio flexible que puede estirarse o encogerse. Por ejemplo, si tienes dos bloques de texto y un espacio entre ellos que has definido con alguna cantidad de pegamento, TeX puede decidir aumentar o disminuir este espacio para lograr un diseño de página óptimo.

```
\hbox{Bloque de texto 1}\hskip 1plus.5minus.2\hbox{Bloque de texto 2}
```

2. En un kern que es seguido inmediatamente por pegamento: Un kern en TeX es un espacio no flexible, a diferencia del pegamento. Por lo tanto, si tienes un kern seguido por pegamento, TeX tiene la opción de hacer un salto de página allí.

```
\hbox{Bloque de texto 1}\kern1in\hskip1in\hbox{Bloque de texto 2}
```

3. En una penalidad: Una penalidad en TeX es una forma de indicar lugares preferibles o desfavorables para los saltos de página o de línea. Puedes insertar una penalidad para influir en el lugar donde TeX decidirá hacer el salto de página.

```
\hbox{Bloque de texto 1}\penalty-100\hbox{Bloque de texto 2}
```

Penalización de punto de roptura

Las penalidades en TeX sirven para controlar el comportamiento de los saltos de página y de línea. TeX incluye varias penalidades predefinidas que puedes ajustar según tus necesidades:

1. `\interlinepenalty`: Penalidad por romper una página entre líneas de un párrafo. Un valor alto hará menos probable un salto de página aquí.
2. `\clubpenalty`: Penalidad adicional por romper una página después de la primera línea de un párrafo. Un valor alto hace menos probable que la primera línea de un párrafo aparezca sola al final de una página (lo que se conoce como línea huérfana).
3. `\widowpenalty`: Penalidad adicional por romper una página antes de la última línea de un párrafo. Un valor alto hace menos probable que la última línea de un párrafo aparezca sola al principio de una página (lo que se conoce como línea viuda).
4. `\displaywidowpenalty`: Penalidad adicional por romper una página antes de la última línea antes de una fórmula de visualización.
5. `\brokenpenalty`: Penalidad adicional por romper una página después de una línea con guiones. Un valor alto hace menos probable un salto de página después de una línea que termina con un guión.

Computacion de punto de roptura

Cuando TeX está procesando un documento, se encuentra con diferentes tipos de elementos como cajas, pegamento y kerns. Cada vez que se mueve un elemento a la página actual, TeX calcula la penalidad y la mala calidad (badness en inglés) asociada con romper la página en ese lugar.

La penalidad es la que hemos discutido anteriormente, mientras que la mala calidad es una medida de cuánto tendría que estirarse o encogerse el pegamento en la página para hacer que todo encaje perfectamente.

En base a la penalidad y la mala calidad, TeX calcula un costo total asociado a hacer un salto de página en ese lugar. El lugar de menor costo se recuerda, y cuando el costo es infinito (es decir, la página está demasiado llena), o cuando la penalidad es muy alta (es decir, hay un lugar muy desfavorable para un salto de página), se rompe la página en el lugar de menor costo recordado.

Estos cálculos son esenciales para la forma en que TeX decide dónde hacer saltos de página, lo que a su vez afecta a la apariencia global de los documentos producidos con TeX. Por lo tanto, tener una comprensión de cómo funcionan puede ser muy útil para controlar la disposición de tus propios documentos.

14.22 Rutinas de salida

1 `\output`: TeX utiliza la rutina de salida definida por el token list `\output` para procesar las páginas generadas. Ejemplo:

```
\output={\shipout\box255}
```

2 `\shipout`: Esta instrucción envía una caja a la salida (generalmente un archivo DVI o PDF). Ejemplo:

```
\setbox0=\hbox{Hello World!}
\shipout\box0
\end
```

3 `\mark`: Esta instrucción se utiliza para marcar posiciones en el texto que luego pueden ser referenciadas por la rutina de salida. Ejemplo:

```
\output={\immediate\write16{The last mark was \botmark}}
Hello\mark{1} World\mark{2}
\end
```

En este ejemplo, la rutina de salida escribe en la consola el último mark que fue procesado.

4 `\topmark`, `\firstmark`, `\botmark`, `\splitfirstmark`, y `\splitbotmark`: Estos comandos hacen referencia a las marcas establecidas por el comando `\mark` en varias posiciones de la página. `\topmark` es la última marca de la página anterior, `\firstmark` es la primera marca de la página actual, `\botmark` es la última marca de la página actual, `\splitfirstmark` es la primera marca de la página que fue dividida y `\splitbotmark` es la última marca de la página dividida.

5 `\deadcycles` y `\maxdeadcycles`: `\deadcycles` es un contador que lleva la cuenta de cuántas veces se ha llamado a la rutina de salida sin que se realice un `\shipout`. Si `\deadcycles` excede `\maxdeadcycles`, TeX interrumpe el proceso de generación de páginas y emite un error.

6 `\outputpenalty`: Este parámetro tiene el valor de la penalización en el punto de ruptura de página actual. Si la ruptura no fue por una penalización, su valor es 10000.

Rutina de salida se denota por OTR

Introducción

Ahora estudiaremos las "rutinas de salida" (otrs) en TEX y cómo utilizarlas para personalizar la apariencia de los documentos. En esta sección abarcaremos técnicas avanzadas que involucran el uso de marcas, valores de penalización especiales, kerns y cajas especiales, y finalmente, cómo manejar las inserciones en otrs.

En un caja TEX cambia a ciertos modos

Cajas

Una caja es una unidad indivisible de material, dentro de la cual TEX cambia a uno de los modos internos. Por ejemplo, en un `\hbox` TEX está en modo horizontal restringido, lo que significa que los elementos colocados en esa caja se posicionarán uno al lado del otro. Si se compone un `\hbox`, no se dividirá en líneas, sino que se compondrá como una sola línea. De manera similar, los elementos colocados en un `\vbox` se apilan de arriba hacia abajo. Cuando se compone una caja de este tipo, se coloca toda la caja en la misma página.

Listas

Lista: conjunto de cajas o elementos

Una lista se puede entender como un conjunto de cajas (y otros elementos) colocados en posición horizontal o vertical. TEX soporta listas horizontales y verticales. Un ejemplo de lista horizontal es el logotipo de TEX. Sus componentes son las dos letras 'T', 'X', una caja con la letra 'E' y dos elementos de `\kern`. Un ejemplo de lista vertical es una página de texto. Sus componentes son las líneas individuales de texto y las penalizaciones, el pegamento y otros elementos entre ellas.

Bucle principal

Bucle principal prepara páginas para enviar a OTR

También conocido como el Bucle Interno o Control Principal, es donde TEXpasa la mayor parte de su tiempo, preparando páginas de texto que eventualmente se envían al otr. Este bucle consiste en leer caracteres del archivo fuente, escanearlos y convertirlos en tokens, usar los tokens para construir cajas y combinar las cajas en listas.

shipout

Por ejemplo,

```
\shipout\hbox{Hello, world!}
```

Esto creará una página en el archivo DVI o PDF. Por supuesto, nadie quiere usar de esta manera. Su uso principal está en la rutina de salida que se llama cuando TeX decide que tiene suficiente material para expulsar una página completa. La lista vertical principal actual se divide en el salto de página elegido y su contenido se empaqueta, sobre el que puede actuar la rutina de salida.

Otro ejemplo:

```
\shipout\vbox{%
  \makeheadline
  \pagebody
  \makefootline
}
\advancepageno
```

```
    \ifnum\outputpenalty>-\@MM
  \else
    \dosupereject
  \fi
```

Esto envía una pagina completa, adornado con encabezado y pie de de página.

Lista de tokens \output

Se dice comúnmente que "se llama a la rutina de salida" cuando TEXha encontrado un lugar para cortar la lista vertical principal. Sin embargo, en realidad, \output no es una macro sino una lista de tokens que se inserta en el flujo de comandos de TEX.

La inserción de la lista de tokens \output ocurre dentro de un grupo que se abre de forma implícita y TEXentra en modo vertical interno. Debido a este grupo, las asignaciones no locales deben tener el prefijo \global. Además, el modo vertical implica que los espacios son inofensivos durante la ejecución de la rutina de salida.

La lista de tokens \output pertenece a la clase de los parámetros de tokens y funciona de la misma manera que las listas de tokens \toksnnn. Por lo tanto, asignar una rutina de salida puede tener las siguientes formas.

```
  \output<equals><general text>
  \output<equals><filler><token variable>
```

Output y \box 255

La funcionalidad de construcción de páginas en TEX, funciona cortando la página actual en el punto óptimo y guardando todo lo que hay por encima de ese punto en \box255. Luego, los tokens \output se insertan en el flujo de entrada. Cualquier material restante en la lista vertical principal se devuelve a las contribuciones recientes. Si la página se divide en una penalización, ese valor se registra en \outputpenalty y se coloca una penalización de tamaño 10000 en la parte superior de las contribuciones recientes; de lo contrario, \outputpenalty se establece en 10000. Cuando se termina la rutina de salida, se supone que \box255 está vacío. Si no lo está, TEXda un mensaje de error.

Por lo general, la rutina de salida tomará la página, agregará un título y/o una línea inferior y, si es necesario, fusionará algunas inserciones como notas al pie y enviará la página al archivo dvi:

Cuando \box255 llega a la rutina de salida, su altura se ha establecido en \vsize. Sin embargo, el material en ella puede tener una altura considerablemente más pequeña. Por lo tanto, la rutina de salida anterior puede dar lugar a cajas insuficientes. Esto se puede corregir con un \vfil.

La rutina de salida no está obligada a hacer nada útil con \box255; puede vaciarlo o desempaquetarlo para que TEXtenga otra oportunidad de encontrar un corte de página. El número de veces que la rutina de salida aplaza el \shipout se registra en \deadcycles: este parámetro se establece en 0 por \shipout

y se aumenta en 1 justo antes de cada \output. Cuando el número de ciclos muertos alcanza \maxdeadcycles, TEXda un mensaje de error y realiza la rutina de salida predeterminada \shipout\box255 en lugar de la rutina que estaba a punto de comenzar. El formato LATEX tiene un valor mucho más alto para \maxdeadcycles que el TEXplano, porque la rutina de salida en LATEX se llama a menudo para el manejo intermedio de flotantes y notas marginales.

El comando \shipout puede enviar cualquier caja ⟨box⟩ al archivo dvi, no necesariamente tiene que ser la caja 255 o incluso tener que contener la página actual. También no es necesario llamarlo dentro de la rutina de salida.

Si la rutina de salida produce algún material, por ejemplo, llamando a \unvbox255, esto se coloca encima de las contribuciones recientes.

Después de que la rutina de salida finalice, se activa el constructor de páginas. En particular, como la página actual se ha vaciado, se vuelve a leer \vsize. Los cambios realizados en este parámetro dentro de la rutina de salida (usando \global) por lo tanto tendrán efecto.

Marcas

El usuario puede especificar una lista de tokens con

```
\mintinline{tex}{\mark{<text>}}
```

la cual se coloca en un elemento de marca en la lista vertical actual. El texto de la marca está sujeto a expansión, como en \edef.

Si la marca se da en el modo horizontal, se migrará a las listas verticales circundantes como un elemento de inserción; sin embargo, si esta no es la lista vertical externa, la rutina de salida no encontrará la marca.

Las marcas son el mecanismo principal a través del cual la rutina de salida puede obtener información sobre el contenido de la página actualmente rota, en particular su parte superior e inferior. TEXestablece tres variables:

1. \botmark es la última marca que ocurre en la página actual.
2. \firstmark es la primera marca que ocurre en la página actual.
3. \topmark es la última marca de la página anterior, es decir, el valor de \botmark en la página anterior.

Si todavía no ha habido marcas, todas las tres son vacías; si no hay marcas en la página actual, todas las tres variables de marca son iguales a \botmark de la página anterior.

Para las cajas generadas por un comando \vsplit, \splitbotmark y \splitfirstmark contienen las marcas de la parte separada; \firstmark y \botmark reflejan el estado de lo que queda en el registro.

Ejemplo 14.71

```
%page layout
\pdfpagewidth14cm\relax
\pdfpageheight17cm\relax
\vsize13cm\relax
\hsize10cm\relax
%end page layout
\def\section#1{#1\mark{#1}}
\def\rightheadline{\hbox to \hsize{\botmark}}
\def\leftheadline{\hbox to \hsize{\firstmark}}

\section{Test section 1}

\section{Test section 2}

\par\vfil\penalty -10000

\rightheadline
\leftheadline

\bye
```

Test section 1
Test section 2

1

Test section 2
Test section 1

2

La lista vertical principal y el generador de páginas

MVL (main vertical list) es donde se agrega párrafos

Main Vertical List (MVL o lista vertical principal). Cuando TEX lee el primer carácter de un párrafo, cambia a modo horizontal, lee el resto del párrafo y luego cambia de nuevo a modo vertical e invoca el algoritmo de ruptura de línea. Las líneas resultantes se guardan en la MVL y se van agregando más líneas a medida que se leen del archivo de origen. Cuando la MVL contiene suficiente material para una página, TEX invoca al Page Builder (creador de páginas), quien decide dónde cortar la página, corta un trozo de la MVL del tamaño de una página, lo coloca en `\box255` e invoca al otr para agregar elementos como el número de página, encabezados o notas al pie y enviar la página al archivo .dvi.

El OTR toma elementos de MVL y envia a .dvi

`\box 255` es de tamaño `\vsize` y `\hsize`

El otr no tiene que enviar toda la página, puede enviar solo una parte y guardar o descartar el resto, que se devolverá a la MVL. Es importante destacar que el otr no está sincronizado con el bucle principal de TEX, sino que se invoca según sea necesario. TEX no sabe el tamaño del papel que finalmente saldrá de la impresora, solo conoce cuatro parámetros relacionados con el tamaño y la posición de la página: `\hsize`, `\vsize`, `\hoffset` y `\voffset`. Los dos primeros se convierten en el ancho y alto de `\box255`, mientras que los dos últimos son las cantidades por las cuales los márgenes izquierdo y superior difieren de los márgenes predeterminados de la impresora. Los márgenes derecho e inferior son desconocidos para TEX y generalmente solo los usa el controlador de impresora. En casos raros en los que TEX necesita conocer esas cantidades, deben ingresarse o calcularse manualmente.

Pagetotal y pagegoal

`\pagetotal` y `\pagegoal` son utilizadas por el constructor de página y se usan en el algoritmo de roptura de página. Estas variables `\dimen` también son utilizadas en las inserciones en la página.

pagetotal(t) es la altura vertical de MVL

1. La variable `\pagetotal` representa la altura vertical del MVL (Main Vertical List), que se refiere a la lista de cajas y elementos de pegamento que se van acumulando en la página a medida que se procesa el texto. Se denota como "t" y comienza en cero, incrementándose cada vez que se agrega algo con una altura (como una caja o pegamento) al MVL. Elementos como marcas, penalizaciones y whatsits no tienen dimensiones y no afectan a "t". Debido a que algunos pegamentos verticales tienen flexibilidad, esta variable generalmente es flexible. Inicialmente, cuando el MVL está vacío, "t" se establece en cero.

pagegoal(g) es la altura deseada de página

2. Por otro lado, la variable `\pagegoal`, denotada como "g", representa la altura deseada de la página. Generalmente, es igual a `\vsize` (la altura vertical disponible para el texto), pero cuando se generan inserciones como notas en pie de página "g" se puede decrementar. Inicialmente, cuando el MVL está vacío, "g" se establece en `\maxdimen` (la dimensión máxima posible). Cuando se coloca el primer elemento en el MVL, "g" se establece en `\vsize`.

pagetotal y pagegoal se utiliza para realizar cortado e inserción de páginas

Estas dos variables son importantes para el algoritmo de salto de página en TEX, ya que se utilizan para determinar cuándo se debe agregar una nueva página y cuánto contenido se puede ajustar en cada página. También son útiles para las inserciones en la página, ya que pueden afectar la altura disponible para el texto y las inserciones.

Cuando se generan notas al pie de página u otro material que se va a insertar en la página, TEX debe asegurarse de que este material se ajuste en la página, sin exceder su altura máxima. Para hacerlo, TEX resta la altura de este material de la variable \pagegoal. Esto asegura que se reserve suficiente espacio en la página para el material que se va a insertar.

Estos dos variables pueden ser "mostrados", ya sea en el documento mismo (usando \the) o en el archivo de registro (usando \showthe o \message). Incluso pueden ser modificados, pero esto debe hacerse con mucho cuidado. Incluso es posible obtener los dos valores mostrados después de cada línea al establecer \tracingpages=1. Si estableces \tracingpages=1, podrás ver los valores de \pagegoal y \pagetotal después de cada línea en el archivo de registro. La principal función de esta opción es mostrar los puntos posibles para los saltos de página y cómo funciona el algoritmo de división de páginas. Es recomendable probarlo al menos una vez en una página corta que tenga notas al pie, para observar los cambios en los valores de las dos cantidades.

Ejemplo 14.72

```
\pdfpagewidth450pt \hsize450pt \hoffset-1in
\pdfpageheight100pt \vsize100pt \voffset-1in

Valor de pagetotal: \the\pagetotal

Valor de vsize es \the\vsize y el valor de pagegoal es \the\pagegoal.

Hola mundo 1

Valor de pagetotal después de hola mundo 1: \the\pagetotal

Valor de vsize es \the\vsize y el valor de pagegoal es \the\pagegoal después de hola mundo 1.

Como podemos apreciar el valor de pagetotal aumenta de valor.

\bye
```

Valor de pagetotal: 0.0pt
Valor de vsize es 100.0pty el valor de pagegoal es 100.0pt.
Hola mundo 1
Valor de pagetotal despus de hola mundo 1: 34.0pt
Valor de vsize es 100.0pty el valor de pagegoal es 100.0ptdespus de hola mundo 1.
Como podemos apreciar el valor de pagetotal aumenta de valor.

Como ejemplo, los dos variables se utilizarán para determinar cuánto espacio queda en la página actual. Si t es cero, el espacio restante en la página es toda la página (\vsize). De lo contrario, es la diferencia g-t. La macro \pagespace calcula esa diferencia.

Ejemplo 14.73

```
\pdfpagewidth450pt \hsize450pt \hoffset-1in
\pdfpageheight100pt \vsize100pt \voffset-1in
```

```
\newdimen\spaceleft
\def\pagespace{%
  \ifdim\pagetotal=0pt
    \spaceleft=\vsize
    \the\spaceleft
  \else
    \spaceleft=\pagegoal
    \advance\spaceleft by -\pagetotal
    \the\spaceleft
  \fi}
Espacio que queda: \pagespace

Hola mundo 1

Espacio que queda después de ``Hola mundo 1'' es \pagespace
\bye
```

Espacio que queda: 100.0pt
Hola mundo 1
Espacio que queda despus de “Hola mundo 1” es 78.0pt

La página actual y lista de contribuciones recientes

El MVL (Main Vertical List) se compone de dos partes: la página actual y la lista de contribuciones recientes. La página actual contiene el material que se convertirá en `\box255`, mientras que las contribuciones recientes almacenan temporalmente el material leído recientemente. Después de leer un párrafo completo, este se compone tipográficamente y las líneas de texto se agregan a las contribuciones recientes. Luego, se invoca al constructor de páginas.

El trabajo del constructor de páginas es mover las líneas, una por una, desde las contribuciones recientes a la página actual. Para cada línea, el constructor calcula el costo de dividir la página después de esa línea. Al principio, el costo es muy alto porque resultaría en una página extremadamente estirada. A medida que se agregan suficientes líneas a la página actual, el costo disminuye. Si hay demasiadas líneas en la página actual, se debe reducir su tamaño, aumentando el costo nuevamente. Este proceso puede verse en tiempo real estableciendo `\tracingpages=1`.

El comando `\tracingpages=1` se utiliza en TEX para habilitar el seguimiento de la construcción de páginas, lo que te permite ver cómo se toman las decisiones de división de páginas y cómo se calculan los costos asociados a ellas. Veamos un ejemplo práctico que utiliza `\tracingpages=1` para ilustrar cómo funciona:

Ejemplo 14.74

```
\pdfpagewidth=11cm \hsize=11cm \hoffset=-1in
\pdfpageheight=2cm \vsize=2cm \voffset=-1in
\tracingpages=1 % Habilita el seguimiento de la construcción de páginas
Este es el primer párrafo. Aquí escribimos algunas líneas de texto para mostrar cómo TeX construye la
↪  página. El constructor de páginas se activa al final de un párrafo, al final de una ecuación dentro
↪  de un párrafo, al final de un halign y en algunos otros casos.

Este es el segundo párrafo. Seguimos escribiendo más líneas de texto para ilustrar el proceso de
↪  construcción de páginas en TeX. El otr solo se invoca mediante el constructor de páginas, por lo
↪  que nunca se invoca en medio de un párrafo (a menos que contenga material matemático).
\bye
```

Este es el primer prrafo. Aqu escribimos algunas lneas de texto para mostrar cmo TeX construye la pgina. El constructor de pginas se activa al final de un prrafo, al final de una ecuacin dentro de un prrafo, al final de un halign y en algunos otros casos.

Este es el segundo prrafo. Seguimos escribiendo ms lneas de texto para ilustrar el proceso de construccin de pginas en TeX. El otr solo se invoca mediante el constructor de pginas, por lo que nunca se invoca en medio de un prrafo (a menos que contenga material matemtico).

Al compilar el código anterior, TEX genera el siguiente archivo de registro:

```
%% goal height=56.9055, max depth=4.0
% t=10.0 g=56.9055 b=10000 p=150 c=100000#
% t=22.0 g=56.9055 b=10000 p=0 c=100000#
% t=34.0 g=56.9055 b=10000 p=150 c=100000#
% t=46.0 g=56.9055 b=10000 p=0 c=100000#
% t=58.0 plus 1.0 g=56.9055 b=* p=150 c=*

%% goal height=56.9055, max depth=4.0
% t=10.0 g=56.9055 b=10000 p=150 c=100000#
% t=22.0 g=56.9055 b=10000 p=0 c=100000#
% t=34.0 g=56.9055 b=10000 p=150 c=100000#
% t=46.0 g=56.9055 b=10000 p=0 c=100000#
% t=48.5 plus 1.0fill g=56.9055 b=0 p=-20000 c=-20000#
```

Estas líneas indican lo siguiente:

- t: La altura total acumulada del material en la página actual.
- g: La cantidad de glue en la página actual.
- b: La mala calidad (badness) del ajuste actual de la página, que mide qué tan estirada o comprimida está la página. Un valor de 10000 indica una calidad extremadamente mala.
- p: La penalización asociada con la división de la página en ese punto.
- c: El costo de dividir la página en ese punto, que es una combinación de la mala calidad y la penalización.

El constructor de páginas utiliza estos valores para tomar decisiones sobre dónde dividir la página. En este caso, vemos que la mala calidad disminuye en un punto, y luego aumenta nuevamente, lo que indica que el constructor de páginas está considerando diferentes ubicaciones para dividir la página. En última instancia, el constructor de páginas seleccionará la ubicación de menor costo para dividir la página y crear la salida final.

Si la página debe reducirse más allá de su capacidad máxima, el costo se vuelve infinito. En ese caso, el constructor de páginas retrocede a la línea de texto donde el costo fue más bajo, rompe la parte superior de la página actual en ese punto y la coloca en `\box255`. Luego, devuelve la parte inferior de la página actual a las contribuciones recientes e invoca al output routine (otr).

El constructor de páginas se activa al final de un párrafo, al final de una ecuación en un párrafo, al final de un `\halign` y en algunos otros casos. El otr solo se invoca mediante el constructor de páginas.

El entendimiento del constructor de páginas y el MVL debe incluir también el pegamento (glue) y las penalizaciones (penalties). Al componer tipográficamente un párrafo, las líneas de texto se agregan a las contribuciones recientes con glue y penalizaciones entre ellas. Estos elementos se trasladan a la página actual junto con las líneas de texto. Si la página actual está vacía, todos los glues, kerns y penalizaciones se descartan. Cuando la primera caja se mueve a la página actual, se coloca un glue `\topskip` especial sobre ella para mantener su línea base a una cierta distancia del borde superior de la página.

Cuando se encuentra una penalización menor o igual que -10000, el constructor de páginas divide la página. Si la página resultante no tiene suficientes líneas de texto, puede estar subocupada (underfull). Estas penalizaciones se pueden utilizar para expulsar una página (por ejemplo, con `\vfill\penalty-10000`) o para comunicarse con el otr.

Cabe destacar que una penalización de -10000 no invoca al otr de inmediato. Si se crea dicha penalización dentro de un párrafo, se guarda en las contribuciones recientes junto con las líneas y solo se reconoce como especial cuando el constructor de páginas la mueve a la página actual.

En TEX, una página solo puede dividirse en un pegamento (glue), kern o penalización (penalty). Estos elementos ayudan a determinar cómo se distribuyen y ajustan las líneas de texto en una página.

La profundidad de la página actual

En un documento es necesario que las páginas sucesivas tengan el mismo tamaño vertical similar. La altura de la página se controla por medio del parámetro `\vsize` y como de la misma forma, la profundidad de la caja debe estar controlado por el usuario para obtener mejor apariencia de cada una de las páginas del documento.

En TEX, una caja vertical (`\vbox`) puede contener varias líneas de texto, separadas por pegamento (glue) y penalizaciones (penalties). La profundidad de la `\vbox` es la profundidad del componente inferior. Si este componente es pegamento o una penalización, la profundidad será cero. Sin embargo, si es una caja, su profundidad se convierte en la profundidad de toda la `\vbox`, pero está limitada por el valor del parámetro `\boxmaxdepth`.

Ejemplo 14.75

```
\documentclass{article}
\usepackage[paperwidth=11cm, paperheight=3cm, margin=5mm]{geometry}
\begin{document}
\setlength{\boxmaxdepth}{7pt}
\newbox\mybox
\newbox\tmpboxa
\setbox\mybox\vbox{
  \hbox{Esta es la primera línea.}
  \hbox{Esta es la segunda línea.}
  \hbox{Esta es la tercera línea.}
  \setbox\tmpboxa\hbox{Esta caja tiene profundidad de 5pt}
  \dp\tmpboxa5pt
  \box\tmpboxa
}
\copy\mybox
Altura: \the\ht\mybox, Profundidad: \the\dp\mybox
\end{document}
```

Esta es la primera línea.
Esta es la segunda línea.
Esta es la tercera línea.
Esta caja tiene profundidad de 5pt
Altura: 42.94444pt, Profundidad: 5.0pt

La página actual en TEX es la parte del MVL (Main Vertical List) que contiene el material para `\box255`. La altura actual de la página es "t" (total) y la altura objetivo es "g" (goal). La profundidad de la página es la profundidad de la última línea de texto, que puede variar ligeramente de página a página. Si la última línea de texto contiene un símbolo grande con una profundidad de, por ejemplo, 35pt, el tamaño vertical de la página será `\vsize + 35pt`, lo que hará que la página sea más alta que las demás y estropee la apariencia uniforme del documento.

Para evitar esto, el constructor de páginas utiliza otro parámetro, `\maxdepth`, al agregar líneas a la página actual. En el formato "plain", `\maxdepth` se establece en 4pt. Cuando se agrega una línea con una profundidad de 35pt a la página actual, la profundidad de la página se establece en 4pt y la diferencia de 31pt se suma a su altura "t". Podemos visualizar esto como que la línea base de la página actual ya no coincide con la línea base de la línea inferior, sino que se encuentra 31pt por debajo de ella.

El macro `\pagedepth` (d) contiene la profundidad de la página actual y su valor varían de acuerdo a la línea o pegamento añadido a la página actual. Existen otros macros como `\pagestretch` y `\pageshrink` que representan la cantidad de estiramiento y encogimiento de la página actual.

Controlar la profundidad de la página actual

El valor de `\pagedepth` (d) se limita por `\maxdepth`. Cuando se mueve una linea de texto a la página actual de `profundidad > \maxdepth`, entonces la profundidad de la página actual se establece a `\maxdepth` y su diferencia se incrementa a t. Entonces, la línea base de la página actual se reduce hubicándose por debajo de la línea base del texto añadido.

Ejemplo 14.76

```
\pdfpagewidth=11cm \hsize=3.5in \hoffset=-1in
\pdfpageheight=1.7cm \vsize=1.7cm \voffset=-1in
\tracingpages=1
\font\myfont=cmr10 % Definir una nueva fuente
\myfont % Cambiar a la nueva fuente
Contenido del documento  Contenido del documento Contenido del documento Contenido del documento
↪  Contenido del documento  Contenido del documento  Contenido del documento  Contenido del documento
↪  Contenido del documento  Contenido del documento  Contenido del documento  Contenido del documento
↪  Contenido del documento  Contenido del documento  Contenido del documento
\bye
```

Contenido del documento Contenido del documento Contenido del documento Contenido del documento Contenido del documento Contenido del documento Contenido del documento Contenido del documento Contenido del

documento Contenido del documento Contenido del documento Contenido del documento Contenido del documento Contenido del documento Contenido del documento

Al compilar este documento se obtiene el siguiente registro (en el archivo `.log`):

```
%% goal height=48.36958, max depth=4.0
% t=10.0 g=48.36958 b=10000 p=150 c=100000#
% t=22.0 g=48.36958 b=10000 p=100 c=100000#
% t=34.0 g=48.36958 b=10000 p=0 c=100000#
% t=46.0 g=48.36958 b=10000 p=0 c=100000#
% t=58.0 g=48.36958 b=* p=100 c=*
```

En el registro podemos apreciar que t incrementa de 12 en 12, entonces los espacios entre líneas es de 12pt. La última línea tiene un costo (`c=*`) infinito, por lo que la página se rompe después de la línea 4 y efectivamente en el pdf se rompe en la línea 4. Este registro se representa gráficamente en la figura 14.2(a).

Supongamos que la segunda línea tiene una profundidad de 7pt (aumentado con respecto a los demas líneas), entonces el `\pagedepth` se establece a `\maxdepth=4pt` y la diferencia $7-4=3$ se suma a t. La línea de base de la página se reduce 3pt debajo de la línea base de la segunda línea (ver figura 14.2(b)). El valor de baselineskip es 12 para ambos casos.

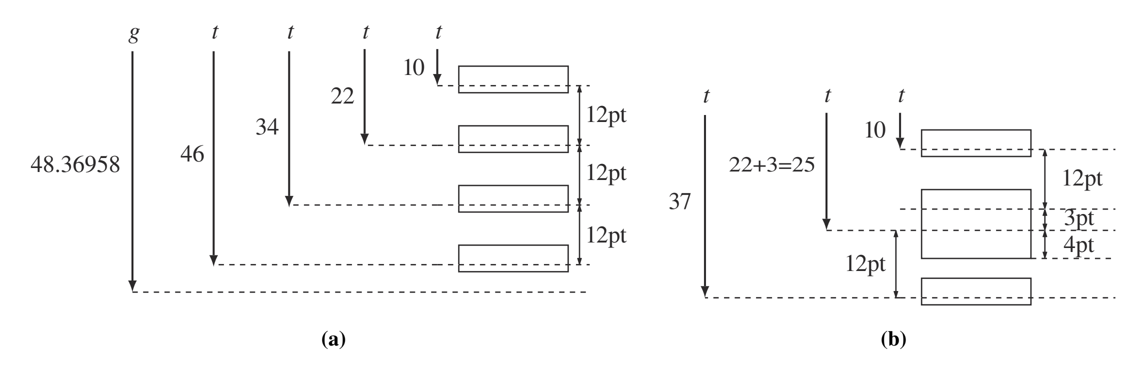


Figura 14.2

La altura de caja de texto

Sea `b=12pt` el valor de `\baselineskip`. Sea `\vbox` que contiene líneas de texto, cada 1 de estas líneas de textos están contenidos en un `\hbox` separados por pegamentos (cantidad de pegamento puede variar) de tal modo que las líneas de base estén separados por una cantidad `b`. Entonces la altura de la caja vertical es

```
b(n-1) + altura de primera línea
```

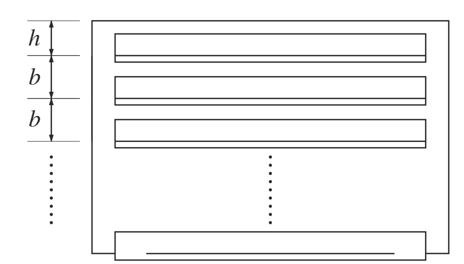
Donde `n` es el número de líneas de texto.

La altura de box255

Cuando se coloca la primera línea de texto en `\box`255, se agrega suficiente pegamento (`\glue`) por encima de ella para alcanzar la distancia especificada por `\topskip` desde la primera línea base. Denotamos el valor de `\topskip` por `h`.

Entonces, si la línea base de la primera línea está ahora `h` (10pt) por debajo de la parte superior de la página, la altura `H` de `\box`255 debe ser `b(n-1) + h`, como se muestra en la figura 14.3.

Figura 14.3



Sin embargo, el constructor de la página siempre establece la altura de `\box`255 en `\vsize`. La diferencia entre las dos alturas suele ser proporcionada por pegamentos flexibles en la página, siendo el más común `\parskip`.

Ejemplo 14.77

```
\pdfpagewidth450pt \hsize450pt \hoffset-1in
\pdfpageheight50pt \vsize50pt \voffset-1in
Contenido del documento Contenido del documento Contenido del documento Contenido del documento
↪ Contenido del documento Contenido del documento Contenido del documento Contenido del documento
↪ Contenido del documento Contenido del documento Contenido del documento Contenido del documento
↪ Contenido del documento Contenido del documento Contenido del documento
\bye
```

Contenido del documento Contenido del documento Contenido del documento Contenido del documento Contenido del documento Contenido del documento Contenido del documento Contenido del documento Contenido del documento Contenido del documento Contenido del documento Contenido del documento Contenido del documento Contenido del documento Contenido del documento

En este ejemplo el valor de `\vsize` es 50pt y se genera 4 líneas de texto, ya que

$$b(n-1)+h = \text{vsize}$$

$$12(4-1)+10 = \text{vsize}$$

$$46 = \text{vsize}$$

Recordar: no importa la capacidad de `\box 255` para empaquetarse y pasar al otr

Entonces el valor del pegamento `\parskip` se estiraron $50-46=4$pt para llenar la página.

Si ya no hay suficiente capacidad de estiramiento, entonces en la parte inferior de la página suele dejarse vacío. .

Ejemplo 14.78

```
\pdfpagewidth400pt \hsize400pt \hoffset-1in
\pdfpageheight100pt \vsize100pt \voffset-1in
\parskip0pt
\output={%
  \setbox0=\vbox{\unvcopy255}
  \message{[\the\ht0, \the\ht255]}
  \setbox1=\vbox to \vsize{\hrule width3in \vfil\hrule width3in} \wd1=0pt
  \shipout\hbox{\box1\box255}
  \advancepageno}
Contenido del documento Contenido del documento Contenido del documento Contenido del documento
↪ Contenido del documento Contenido del documento Contenido del documento Contenido del documento
↪ Contenido del documento Contenido del documento Contenido del documento Contenido del documento
↪ Contenido del documento Contenido del documento Contenido del documento
\bye
```

Contenido del documento Contenido del documento Contenido del documento Contenido del documento Contenido del documento Contenido del documento Contenido del documento Contenido del documento Contenido del documento Contenido del documento Contenido del documento Contenido del documento Contenido del documento Contenido del documento Contenido del documento

Dentro de la rutina de salida \output, se crea un vbox (caja vertical) en el registro 0 utilizando el contenido de la página actual en el registro 255. Se envía las cajas de registros 1 y 255 en un hbox a la salida.

Ejemplos de OTR

Una rutina de salida (otr) es simplemente una cadena de comandos de TEX asignada al registro de tokens \output. Así, al usar \output={...}, TEX ejecutará los comandos "..." cada vez que decida invocar la rutina de salida. La rutina de salida más básica es \output={}. Al encontrar esta rutina de salida, TEX la reemplaza por la predeterminada, que es: \output={\shipout\box255}. Esta rutina de salida predeterminada es la forma más sencilla de enviar una página.

\shipout es un comando primitivo de TEX que genera una página en el archivo DVI. Dicha página refleja el contenido de cualquier caja que siga a la llamada a \shipout. Un aspecto interesante es que \shipout se puede utilizar en cualquier momento, no solo desde una rutina de salida.

Además, cabe destacar que \shipout puede redefinirse (lo cual es aplicable a cualquier secuencia de control, ya sea una macro o un comando primitivo).

Existe la posibilidad de redefinir la rutina de salida en un documento TEX esta rutina definida se ejecutará cuando el constructor de página invoque la próxima vez al otr. La definición podría ser de la siguiente manera \def\newotr{} y luego este macro podemos asignar de la forma \output={\newotr}.

```
\pdfpagewidth400pt \hsize400pt \hoffset-1in
\pdfpageheight100pt \vsize80pt \voffset-1in
\input plipsum
\output={%
  \shipout\box255 \advancepageno
  \global\output={%
    \shipout\vbox{%
      \box255
      \centerline{\folio}
    }
    \advancepageno
  }
}
\lipsum{1-3}
\bye
```

En este ejemplo, la rutina de salida inicialmente definida realiza las siguientes acciones:

1. Envía la caja 255 con \shipout\box255.
2. Incrementa el número de página con \advancepageno.
3. Redefine globalmente la rutina de salida a una nueva definición. La nueva rutina de salida hace lo siguiente:

(a) Envía una caja vertical que contiene:
   i. La caja 255 con el contenido de la página.
   ii. El número de página centrado con \centerline{\folio}.
(b) Incrementa el número de página con \advancepageno.

De esta manera, la primera página se enviará usando la rutina de salida inicial, mientras que las páginas siguientes se enviarán usando la nueva rutina de salida, que incluye el número de página centrado debajo del contenido de la página.

El otr espera que \box255 se vacie, mueva a otra caja o se devuelva a MVL. Si no se hace nada con la caja 255 se provoca un mensaje de error "unvoid \box255".

El código \output={\uvbox255} (otr) devuelve la caja 255 o página a MVL, haciendo que el constructur busque y encuentre un nuevo salto de página y luego invoque nuevamente al otr. Es importante tener en cuenta que el nuevo salto de página podría no ser el mismo que el original debido a la penalización en el punto de interrupción.

Cuando el constructor de páginas elige un punto de interrupción, coloca la penalización encontrada en la variable \outputpenalty, no en \box255. La otr puede devolver la penalización a su lugar original usando \unvbox255\penalty\outputpenalty. Esto garantiza que el constructor de páginas encuentre el mismo punto de interrupción.

Cuando el otr se ejecuta y no envía ninguna página, se la denomina ciclo muerte (dead cycle) y hay que tratar de evitar este ciclo, aunque este ciclo muerto tiene algunas aplicaciones.

Cuando un documento tiene muchos ciclos muertos consecutivos, normalmente indica un error en el código. Para prevenir que estos errores generen un bucle infinito, TEXlleva la cuenta de los ciclos muertos consecutivos en el registro \deadcycles. Si \deadcycles es igual o mayor a \maxdeadcycles, TEX detiene la ejecución. En el formato plano o plain, el valor de \maxdeadcycles es 25, pero se puede modificar en cualquier momento. Cada vez que se invoca \shipout, se reinicia el contador \deadcycles.

El número de página

Ahora nos enfocamos en la personalización del número de páginas.

Primero, se crea una nueva variable \count llamada \pageNum. Luego definimos la rutina de salida de tal modo que coloquen los números de página en el área de impresión.

Ejemplo 14.79

```
\pdfpagewidth400pt \hsize400pt \hoffset-1in
\pdfpageheight100pt \vsize80pt \voffset-1in
\input plipsum
\output={%
  \shipout\box255 \advancepageno
  \global\output={%
    \shipout\vbox{%
```

```
      \box255
      \centerline{\folio}
    }
    \advancepageno
  }
}
\lipsum{1-3}
\bye
```

Stoici autem, quod finem bonorum in una virtute ponunt, similes sunt illorum; Erat enim Polemonis. Ut optime, secundum naturam affectum esse possit. Negat esse eam, inquit, propter se expetendam. Beatus autem esse in maximarum rerum timore nemo potest. Idem iste, inquam, de voluptate quid sentit? Non est ista, inquam, Piso, magna dissensio.

Non ergo Epicurus ineruditus, sed ii indocti, qui, quae pueros non didicisse turpe est, ea putant usque ad senectutem esse discenda. Hoc ille tuus non vult omnibusque ex rebus voluptatem quasi mercedem exigit. Etsi ea quidem, quae adhuc dixisti, quamvis ad aetatem recte isto modo dicerentur. Nemo nostrum istius generis asotos iucunde putat vivere.

Quorum sine causa fieri nihil putandum est. A primo, ut opinor, animantium ortu petitur origo summi boni. Ita enim vivunt quidam, ut eorum vita refellatur oratio. Eam tum adesse, cum dolor omnis absit; Verba tu fingas et ea dicas, quae non sentias? His singulis copiose responderi solet, sed quae perspicua sunt longa esse non debent.

2

La rutina de salida realiza las siguientes acciones:

1. Envía el contenido de `\box255` (la página actual) al archivo DVI.
2. Añade un pequeño espacio vertical (`\smallskip`).
3. Centra el número de página actual (`\centerline{\tenrm\the\pageNum}`) utilizando la fuente `\tenrm` para garantizar que siempre se utilice la misma fuente, independientemente de la fuente actual en el documento.
4. Incrementa el número de página (`\global\advance\pageNum by1`).

Tomar en cuenta los siguientes puntos:

- `\folio`: Muestra el valor de `\count0` como el número de página. Si `\count0` es negativo, `\folio` muestra un numeral romano.
- `\advancepageno`: Incrementa el número de página en uno. Esto se hace incrementando o disminuyendo `\count0` según su signo.

Es posible utilizar cualquier variable `\count0` para contar número de páginas, aunque la ventaja de usar `\count0` es que TeX escribe su valor en el archivo DVI, lo que facilita la visualización del número de página con la página en un programa de vista previa.

Agrupación de OTR

Como la rutina de salida es una lista de tokens agrupados con llaves (grupo), entonces las acciones dentro de esta llave son locales y para que afecte a todo el documento se debe utilizar `\global` dentro del grupo de rutinas de salida.

Otros ejemplos

Página en caja

Ejemplo 14.80

```
\pdfpagewidth426pt \hsize400pt \hoffset-1in
\pdfpageheight140pt \vsize80pt \voffset-1in
\input plipsum
\def\boxit#1#2{%
  \vbox{\hrule
    \hbox{%
      \vrule \kern#2pt
      \vbox{\kern#2pt #1\kern#2pt}%
      \kern#2pt\vrule
    }\hrule
  }
}
\output={
  \shipout\vbox{
    \boxit{\boxit{\box255}9}3
    \medskip
    \centerline{\tenrm\folio}
  }
  \advancepageno
}
\lipsum{1-3}
\bye
```

Stoici autem, quod finem bonorum in una virtute ponunt, similes sunt illorum; Erat enim Polemonis. Ut optime, secundum naturam affectum esse possit. Negat esse eam, inquit, propter se expetendam. Beatus autem esse in maximarum rerum timore nemo potest. Idem iste, inquam, de voluptate quid sentit? Non est ista, inquam, Piso, magna dissensio.

Non ergo Epicurus ineruditus, sed ii indocti, qui, quae pueros non didicisse turpe est, ea putant usque ad senectutem esse discenda. Hoc ille tuus non vult omnibusque ex rebus

1

voluptatem quasi mercedem exigit. Etsi ea quidem, quae adhuc dixisti, quamvis ad aetatem recte isto modo dicerentur. Nemo nostrum istius generis asotos iucunde putat vivere.

Quorum sine causa fieri nihil putandum est. A primo, ut opinor, animantium ortu petitur origo summi boni. Ita enim vivunt quidam, ut eorum vita refellatur oratio. Eam tum adesse, cum dolor omnis absit; Verba tu fingas et ea dicas, quae non sentias? His singulis copiose responderi solet, sed quae perspicua sunt longa esse non debent.

2

Este ejemplo crea la caja `\box255` (página en caja) enmarcado con líneas. El macro `\boxit` recibe dos parámetros, una caja y el espacio entre líneas y texto.

Header y footer

Ahora generaremos un encabezado y un pie de página personalizado con las rutina de salida. El encabezado y el pie de página será una lista de tokens.

Ejemplo 14.81

```
\pdfpagewidth400pt \hsize400pt \hoffset-1in
\pdfpageheight120pt \vsize80pt \voffset-1in
\input plipsum
\output={
  \shipout\vbox{
    \offinterlineskip
    \vbox to1pc{\line{\the\headline}\vss}
    \box255
    \vbox to1pc{\vss\line{\the\footline}}}
  \advancepageno}
\headline={%
  \ifodd\pageno
    \line{\hfil\bf Encabezado\hfil
    \llap{\tenrm\folio}}%
  \else
    \line{\rlap{\tenrm\folio}\hfil
    \bf Encabezado\hfil}%
  \fi}
\footline={\bf Pie de texto\hfil}
\lipsum{1-3}
\bye
```

Encabezado 1

Stoici autem, quod finem bonorum in una virtute ponunt, similes sunt illorum; Erat enim Polemonis. Ut optime, secundum naturam affectum esse possit. Negat esse eam, inquit, propter se expetendam. Beatus autem esse in maximarum rerum timore nemo potest. Idem iste, inquam, de voluptate quid sentit? Non est ista, inquam, Piso, magna dissensio.

Non ergo Epicurus ineruditus, sed ii indocti, qui, quae pueros non didicisse turpe est, ea putant usque ad senectutem esse discenda. Hoc ille tuus non vult omnibusque ex rebus

Pie de texto

2 **Encabezado**

voluptatem quasi mercedem exigit. Etsi ea quidem, quae adhuc dixisti, quamvis ad aetatem recte isto modo dicerentur. Nemo nostrum istius generis asotos iucunde putat vivere.

Quorum sine causa fieri nihil putandum est. A primo, ut opinor, animantium ortu petitur origo summi boni. Ita enim vivunt quidam, ut eorum vita refellatur oratio. Eam tum adesse, cum dolor omnis absit; Verba tu fingas et ea dicas, quae non sentias? His singulis copiose responderi solet, sed quae perspicua sunt longa esse non debent.

Pie de texto

La rutina de salida `\output` comienza enviando una caja vertical que contiene el encabezado, el cuerpo de la página y el pie de página. El encabezado y el pie de página se configuran para ocupar 1pc cada uno.

El encabezado (`\headline`) y el pie de página (`\footline`) son listas de tokens, lo que significa que pueden contener cualquier secuencia de comandos TEX, incluyendo condicionales como `\if`.

Una página de título para un capítulo

Ejemplo 14.82

```
\pdfpagewidth400pt \hsize400pt \hoffset-1in
\pdfpageheight80pt \vsize80pt \voffset-1in
\input plipsum
\def\chapter#1#2;{%
  \vfill\eject
  \shipout\vbox to\vsize{%
```

```
    \line{\bf Capitulo\hfil#1}
    \vfil\vbox{\bf#2}\vfil}
  \advancepageno}
\chapter 1 VECTORES;
\lipsum{1}
\bye
```

Capitulo 1

VECTORES

Stoici autem, quod finem bonorum in una virtute ponunt, similes sunt illorum; Erat enim Polemonis. Ut optime, secundum naturam affectum esse possit. Negat esse eam, inquit, propter se expetendam. Beatus autem esse in maximarum rerum timore nemo potest. Idem iste, inquam, de voluptate quid sentit? Non est ista, inquam, Piso, magna dissensio.

En el ejemplo se define el comando chapter en donde recibe dos argumentos. El primero representa el número de capítulo y el segundo el título del capítulo.

El comando `\chapter` empieza con `\vfill\eject`. `\vfill` inserta un espacio vertical que se expande para llenar cualquier espacio disponible, lo que empuja cualquier contenido existente hacia arriba. `\eject` fuerza a TEX a terminar la página actual y comenzar una nueva.

Luego, el comando `\chapter` invoca `\shipout\vbox to\vsize{...}` envía la caja vertical al dvi y finalmente avanza de página.

Al momento de invocar el comando `\chapter` puede surgir dos problemas.

1. Si el comando `\chapter` se invoca en la primera página, entonces generaría una página en blanco, pero esto no sucede.
2. Si una página fue expulsado justo antes de invocar este comando, entonces podría provocar que se expulse otra página en blanco.

Este es dos problemas no son reales debido a que el comando `\eject` es una penalización de -10000. Si en la parte superior de la página es una penalización, entonces se obvia o descarta esta penalización.

Un borde alrededor de la página

Ejemplo 14.83

```
\pdfpagewidth420pt \hsize400pt \hoffset-1in
\pdfpageheight100pt \vsize80pt \voffset-1in
\input plipsum
\setbox2=\vbox to3.5in{
  \offinterlineskip
  \hbox to 420pt{\leaders\hbox{a}\hfill}
  \hbox to 420pt{%
    \leaders
    \vbox to 85pt{
      \leaders
```

```
      \hbox to 420pt{m\hfil n}
      \vfil}%
    \hfil}
    \hbox to 420pt{\leaders\hbox{b}\hfill}
    \vss}
\wd2=0pt \ht2=0pt \dp2=0pt
\output={
  \shipout\vbox{
    \copy2
    \vskip5pt
    \moveright10pt\box255}
  \advancepageno}
\lipsum{1-3}
\bye
```

Stoici autem, quod finem bonorum in una virtute ponunt, similes sunt illorum; Erat enim Polemonis. Ut optime, secundum naturam affectum esse possit. Negat esse eam, inquit, propter se expetendam. Beatus autem esse in maximarum rerum timore nemo potest. Idem iste, inquam, de voluptate quid sentit? Non est ista, inquam, Piso, magna dissensio.

Non ergo Epicurus ineruditus, sed ii indocti, qui, quae pueros non didicisse turpe est, ea putant usque ad senectutem esse discenda. Hoc ille tuus non vult omnibusque ex rebus

voluptatem quasi mercedem exigit. Etsi ea quidem, quae adhuc dixisti, quamvis ad aetatem recte isto modo dicerentur. Nemo nostrum istius generis asotos iucunde putat vivere.

Quorum sine causa fieri nihil putandum est. A primo, ut opinor, animantium ortu petitur origo summi boni. Ita enim vivunt quidam, ut eorum vita refellatur oratio. Eam tum adesse, cum dolor omnis absit; Verba tu fingas et ea dicas, quae non sentias? His singulis copiose responderi solet, sed quae perspicua sunt longa esse non debent.

14.23 Inserciones

Las inserciones son la forma en que TEXmaneja la información flotante. El constructor de páginas de TEXcalcula qué inserciones y cuántas de ellas caben en la página; luego, estos elementos de inserción se colocan en cajas de inserción que deben ser manejadas por la rutina de salida.

1. `\insert`: Inicia un elemento de inserción.
2. `\newinsert`: Asigna una nueva clase de inserción.
3. `\insertpenalties`: Total de penalizaciones para inserciones divididas. Dentro de la rutina de salida, el número de inserciones retenidas.
4. `\floatingpenalty`: Penalización agregada cuando una inserción se divide.
5. `\holdinginserts` (solo TEX3): Si es positivo, las inserciones no se colocan en sus cajas en el momento de la salida.
6. `\footins`: Número de la clase de inserción de notas al pie en TEXpuro.
7. `\topins`: Número de la clase de inserción superior.
8. `\topinsert`: Macro de TEXpuro para iniciar una inserción superior.
9. `\pageinsert`: Macro de TEXpuro para iniciar una inserción que ocupará una página completa.
10. `\midinsert`: Macro de TEXpuro que coloca su argumento si hay espacio y lo convierte en una inserción superior en otro caso.

11 `\endinsert`: Macro de TEXpuro para finalizar un elemento de inserción que comenzó con `\topinsert`, `\midinsert` o `\pageinsert`.

Topinsert, midinsert y botinsert

Estos comandos producen diferentes formas de inserción que indican (o permiten) a TEXmover el ⟨vertical mode material⟩:

`\topinsert` intenta colocar el material en la parte superior de la página actual. Si no caben allí, `\topinsert` lo moverá al siguiente lugar disponible en la parte superior de la página.

`\midinsert` intenta colocar el material en la posición actual. Si no caben allí, `\midinsert` lo moverá al siguiente lugar disponible en la parte superior de la página.

`\pageinsert` coloca el material solo en la siguiente página. Para evitar una página incompleta, asegúrese de finalizar el material insertado con `\vfil` o rellene el espacio restante de otra manera.

Se dice que el ⟨vertical mode material⟩ es "flotante" porque TEXpuede moverlo de un lugar a otro. Las inserciones son muy útiles para materiales como figuras y tablas, ya que puede posicionar ese material donde desee sin saber dónde caerán los saltos de página.

Cada uno de estos comandos termina implícitamente el párrafo actual, por lo que solo debe usarlos entre párrafos. No debe usarlos dentro de una caja o dentro de otra inserción. Si tiene varias inserciones compitiendo por el mismo espacio, TEXmantendrá su orden relativo.

14.24 Entrada y salida de archivos

1 `\input`: Lee un archivo específico como entrada de TEX.

Este comando permite incluir el contenido de otro archivo en el archivo actual. Por ejemplo, si tienes un archivo llamado "texto.tex" que contiene texto que deseas incluir en tu documento actual, puedes usar `\input{texto.tex}` para insertar ese contenido en el lugar deseado.

2 `\endinput`: Termina la entrada del archivo actual después de la línea actual.

Este comando se utiliza para detener la lectura de un archivo antes de que se llegue al final. Puede ser útil cuando se desea interrumpir la lectura de un archivo antes de que se complete por completo.

3 `\pausing`: Especifica que TEX debe pausar después de cada línea leída desde un archivo.

Al usar este comando, TEX mostrará cada línea leída desde un archivo en la consola o en el archivo de registro, lo que puede ser útil para propósitos de depuración o seguimiento del proceso de lectura.

4 `\inputlineno`: Devuelve el número de la línea actual de entrada.

Este comando se utiliza para obtener el número de línea actual durante la lectura de un archivo. Puede ser útil para realizar un seguimiento de la posición actual dentro del archivo o para fines de referencia.

5 `\write`: Escribe un ⟨general text⟩ en la terminal o en un archivo.

Este comando se utiliza para escribir texto en la consola o en un archivo. Permite generar salidas personalizadas durante la ejecución del programa TEX. Por ejemplo, puedes usar `\write\outputfile{Hello, World!}` para escribir la cadena "Hello, World!" en el archivo especificado por `\outputfile`.

6 `\read`: Lee una línea de un flujo y la asigna a una secuencia de control.

Este comando se utiliza para leer una línea de un flujo de entrada y asignarla a una secuencia de control específica. Por ejemplo, puedes usar `\read\inputfile to \myvariable` para leer una línea del archivo especificado por `\inputfile` y almacenarla en la secuencia de control `\myvariable`.

7 `\newread` y `\newwrite`: Macro para asignar un nuevo flujo de entrada/salida.

Estas macros se utilizan para asignar nuevos flujos de entrada o salida. Por ejemplo, puedes usar `\newread\inputfile` para asignar un nuevo flujo de entrada llamado `\inputfile`. Posteriormente, puedes utilizar `\openin\inputfile=filename` para abrir un archivo específico en ese flujo de entrada.

8 `\openin` y `\closein`: Abre y cierra un flujo de entrada.

Estos comandos se utilizan para abrir y cerrar flujos de entrada. Por ejemplo, puedes usar `\openin\inputfile=filename` para abrir un archivo específico en el flujo de entrada `\inputfile`, y luego puedes usar `\closein\inputfile` para cerrar ese flujo de entrada.

9 `\openout` y `\closeout`: Abre y cierra un flujo de salida.

Estos comandos se utilizan para abrir y cerrar flujos de salida. Por ejemplo, puedes usar `\openout\outputfile=filename` para abrir un archivo específico en el flujo de salida `\outputfile`, y luego puedes usar `\closeout\outputfile` para cerrar ese flujo de salida.

10 `\ifeof`: Prueba si se ha leído por completo un archivo o si no existe.

Este comando se utiliza para verificar si un archivo ha sido leído completamente o si no existe. Puedes usarlo en combinación con otros comandos, como `\openin` y `\closein`, para realizar acciones específicas dependiendo del estado del archivo.

11 `\immediate`: Prefijo para ejecutar operaciones de salida de inmediato.

Este prefijo se utiliza para asegurar que las operaciones de salida se ejecuten de inmediato en lugar de esperar a que TEX decida cuándo realizar las operaciones de salida. Por ejemplo, puedes usar `\immediate\write\outputfile{Hello, World!}` para escribir la cadena "Hello, World!" en el archivo especificado por `\outputfile` de inmediato.

12 `\escapechar`: Número del carácter utilizado cuando las secuencias de control se convierten en tokens de caracteres.

Este comando se utiliza para especificar el número del carácter que se utiliza cuando TEX convierte secuencias de control en tokens de caracteres. Por defecto, este número es 92, que corresponde al carácter de barra invertida "".

13 `\newlinechar`: Número del carácter que provoca un salto de línea en las declaraciones `\write`.

Este comando se utiliza para especificar el número del carácter que provoca un salto de línea en las declaraciones `\write`. Por ejemplo, puedes usar `\newlinechar=10` para especificar que el carácter de nueva línea es el número 10, que corresponde al carácter de retorno de carro "`\n`" . Cuando se utiliza `\write`, TEX realizará un salto de línea cuando encuentre ese carácter.

Including files

En TEX, se pueden segmentar documentos grandes dividiéndolos en archivos separados y cargándolos en el archivo principal utilizando el comando `\input`. La sintaxis exacta para los nombres de archivo depende de la implementación; la mayoría de las veces se asume la extensión .tex si no se proporciona una extensión explícita. Los nombres de archivo pueden estar delimitados por un espacio o por `\relax`. El comando `\input` es expansible, lo que significa que puede ser utilizado en contextos donde se esperan expansiones de comandos.

Cuando TEX encuentra en un archivo de entrada la instrucción `\endinput`, actúa como si el archivo terminara después de la línea en la que se encuentra la instrucción. Sin embargo, cualquier instrucción en la misma línea que `\endinput` aún se ejecuta. La instrucción `\endinput` también es expansible.

Supongamos que tenemos un archivo llamado "contenido.tex" que contiene el siguiente contenido:

```
Este es el contenido de un archivo de ejemplo.
```

En el archivo principal, podemos usar el comando `\input` para incluir el contenido del archivo "contenido.tex" de la siguiente manera:

```
\input contenido.tex
```

Esto insertará el contenido del archivo "contenido.tex" en el lugar donde se encuentra el comando `\input`.

Supongamos que tenemos un archivo llamado "configuracion.tex" que contiene la configuración de nuestro documento, y queremos detener la inclusión de más contenido después de esa configuración. Podemos lograrlo utilizando el comando `\endinput` de la siguiente manera:

```
\input configuracion.tex
\endinput
```

En este caso, cualquier contenido que esté después del comando `\endinput` en el archivo principal no se incluirá en el documento final. Sin embargo, las instrucciones en la misma línea que `\endinput` aún se ejecutarán.

File I/O

TEX admite la lectura y escritura de archivos línea por línea a través de flujos de entrada y salida. Estos flujos permiten interactuar con archivos externos desde dentro de un documento TEX. A continuación, se explicarán los conceptos clave relacionados con la entrada y salida de archivos.

Flujos de apertura y cierre

TEX admite hasta 16 flujos de entrada y 16 flujos de salida simultáneos. Los macros básicos de TEX, como los de Plain TEX, como `\newread` y `\newwrite`, permiten asignar un número a un flujo no utilizado. Este número se asigna mediante el comando `\chardef`. Los flujos de entrada y salida son completamente independientes entre sí.

Los flujos de entrada se abren mediante el comando:

```
\openin<número de 4 bits><signo igual><nombre de archivo>
```

Y se cierran mediante el comando:

```
\closein⟨número de 4 bits⟩
```

Los flujos de salida se abren mediante el comando:

```
\openout<número de 4 bits><signo igual><nombre de archivo>
```

Y se cierran mediante el comando:

```
\closeout<número de 4 bits>
```

Si un archivo de salida no existe, se crea al utilizar `\openout`. Si el archivo ya existía, al utilizar `\openout` se sobrescribirá.

Operaciones de salida con `\openout`, `\closeout` y `\write` también pueden ser precedidas por `\immed⌋ iate`, lo cual se explicará más adelante.

Supongamos que queremos abrir el archivo "datos.txt" para lectura y asignarle el flujo de entrada número 3. Podemos hacerlo de la siguiente manera:

```
\newread\myinput
\openin\myinput=datos.txt
```

En este ejemplo, el flujo de entrada se ha asignado a la variable `\myinput`, y el archivo "datos.txt" se ha abierto para su lectura.

Supongamos que queremos abrir el archivo "resultado.txt" para escritura y asignarle el flujo de salida número 1. Podemos hacerlo de la siguiente manera:

```
\newwrite\myoutput
\openout\myoutput=resultado.txt
```

En este ejemplo, el flujo de salida se ha asignado a la variable \myoutput, y el archivo "resultado.txt" se ha abierto para su escritura.

Entrada con read

Además del comando \input, que lee un archivo completo, TEX cuenta con la operación \read, que lee una línea de un archivo (o desde la terminal del usuario). La sintaxis del comando \read es la siguiente:

```
\read<número>to<secuencia de control>
```

El efecto de esta declaración es que se lee una línea de entrada del flujo designado y la secuencia de control se define como una macro sin parámetros, teniendo esa línea como texto de reemplazo.

Si la línea de entrada no está balanceada con respecto a las llaves, TEX leerá más de una línea, continuando hasta obtener una lista de tokens balanceada. TEX añade implícitamente una línea vacía a cada flujo de entrada, por lo que la última operación \read en un flujo siempre producirá un único token \par.

Las operaciones de lectura desde cualquier flujo fuera del rango de 0 a 15, o flujos no asociados con un archivo abierto, o en los que se ha alcanzado el final del archivo, leen desde la terminal. Si el número del flujo es positivo, se le muestra al usuario el nombre de la secuencia de control que se está definiendo mediante la instrucción \read.

Supongamos que queremos leer una línea de un archivo y asignarla a la macro \data. Podemos hacerlo de la siguiente manera:

```
\newread\myinput
\openin\myinput=datos.txt
\read\myinput to\data
```

En este ejemplo, se ha abierto el archivo "datos.txt" para su lectura y se ha asignado el flujo de entrada a la variable \myinput. Luego, se utiliza el comando \read para leer una línea del archivo y asignarla a la macro \data.

Supongamos que queremos leer una entrada desde la terminal y asignarla a la macro \inputdata. Podemos hacerlo de la siguiente manera:

```
\read-1 to\inputdata
```

En este ejemplo, el número de flujo -1 indica que la lectura se realizará desde la terminal. La entrada proporcionada por el usuario se asignará a la macro \inputdata.

Salida con write

El comando

```
\write<number><general text>
```

permite escribir una lista de tokens balanceada en un archivo que ha sido abierto con `\openout`, en el archivo de registro o en la terminal.

Las operaciones de escritura en un flujo fuera del rango de 0 a 15, o en un flujo que no está asociado con un archivo abierto, se envían al archivo de registro. Si el número del flujo es positivo, también se envían a la terminal además del archivo de registro.

El argumento de lista de tokens de `\write`, definido como

```
<general text> → <filler>{<balanced text><right brace>
```

puede tener una llave de apertura implícita. Este argumento se expande como si fuera el texto de reemplazo de un `\edef`, lo que significa que se expanden las macros y condicionales que aparezcan. Sin embargo, no se ejecutan comandos. Esta expansión ocurre al momento del envío. Hasta ese momento, la lista de tokens de argumento se almacena en un elemento whatsit en la lista actual.

Una secuencia de control generada por `\write` (o `\message`) se representa con un espacio al final y se utiliza el número de caracter `\escapechar` como el carácter de escape. El valor predeterminado de `\escapechar` en IniTEX es 92, el código de la barra invertida. El espacio final se puede evitar prefijando la secuencia de control con `\string`.

Supongamos que queremos escribir un mensaje en un archivo de salida llamado "output.txt". Podemos hacerlo de la siguiente manera:

```
\newwrite\myoutput
\openout\myoutput=output.txt
\write\myoutput{Hello, World!}
\closeout\myoutput
```

En este ejemplo, se ha creado un nuevo flujo de salida llamado `\myoutput` y se ha abierto el archivo "output.txt" para escribir en él. Luego, utilizamos el comando `\write` para escribir la frase "Hello, World!" en el archivo. Finalmente, cerramos el flujo de salida con `\closeout`.

Supongamos que queremos mostrar un mensaje en la terminal y escribirlo en el archivo de registro. Podemos hacerlo de la siguiente manera:

```
\write16{This is a message.}
```

En este ejemplo, el número de flujo 16 indica que el mensaje se mostrará en la terminal y se escribirá en el archivo de registro.

whatsit

Hay una diferencia esencial en la ejecución entre entrada y salida: las operaciones relacionadas con la salida (`\openout`, `\closeout`, `\write`) no se ejecutan inmediatamente, sino que se guardan hasta que la caja en la que aparecen se envía al archivo dvi.

Las escrituras y las otras dos operaciones de salida se colocan en elementos "whatsit" en la lista actualmente en construcción. La operación real ocurre cuando se envía la parte de la página que tiene el elemento al archivo dvi. Este retraso en la salida es necesario debido al comportamiento asíncrono de la rutina de salida de TEX. Ver un ejemplo detallado en la página 134.

Una escritura `\immediate\write` o cualquier otra operación de salida `\immediate` se ejecuta en el acto y no coloca un elemento whatsit en la lista actual. El argumento de un comando `\special` también se coloca en un elemento whatsit.

Los elementos whatsit en cajas leaders son ignorados.

Los elementos “whatsit” en TEX son utilizados para diferir la ejecución de operaciones de salida hasta que el bloque correspondiente se envíe al archivo dvi. Esto se debe a la naturaleza asíncrona de la rutina de salida de TEX. Las operaciones de salida que se diferirán incluyen `\openout`, `\closeout` y `\write`.

Cuando se realiza una operación de salida, como `\write`, en lugar de ejecutarse inmediatamente, se guarda en un elemento "whatsit" en la lista que se está construyendo actualmente. Por ejemplo, si usamos `\write` para escribir un mensaje en un archivo de salida, esa operación de escritura se almacena en un elemento "whatsit" en la lista.

La ejecución real de las operaciones de salida diferida ocurre cuando la parte de la página que contiene el elemento “whatsit” se envía al archivo dvi. Es en este momento cuando se realizan las operaciones de salida guardadas en los elementos “whatsit”. Esto permite que las operaciones de salida se realicen en el orden correcto y en el momento adecuado durante el procesamiento del documento.

Por otro lado, existen operaciones de salida que se ejecutan de inmediato sin colocar un elemento “whatsit” en la lista. Por ejemplo, `\immediate\write` se ejecuta de inmediato y no se guarda en un elemento “whatsit”. Esto significa que el resultado de la operación de escritura se muestra o se guarda de inmediato, sin esperar a que se envíe el bloque correspondiente al archivo dvi.

Además, el argumento de un comando `\special` también se coloca en un elemento “whatsit”. Un comando `\special` se utiliza para especificar una instrucción especial para el procesador de salida. Por ejemplo, se puede utilizar un comando `\special` para controlar la impresión de una imagen en el documento.

Es importante tener en cuenta que los elementos “whatsit” en cajas líderes (leader boxes) se ignoran. Los leader boxes son cajas especiales utilizadas en TEX para crear patrones de repetición, como puntos de relleno en una línea.

Aquí tienes algunos ejemplos para ilustrar el concepto de elementos “whatsit” y la diferencia entre la salida inmediata y la salida diferida:

Supongamos que queremos imprimir un mensaje en la terminal y guardar otro mensaje en un archivo de salida. Podemos hacerlo de la siguiente manera:

```
\write16{Message on the terminal.}
\newwrite\myoutput
\openout\myoutput=output.txt
\immediate\write\myoutput{Message in the output file.}
\closeout\myoutput
```

En este ejemplo, utilizamos `\write16` para imprimir el mensaje "Message on the terminal" directamente en la terminal. Luego, creamos un nuevo flujo de salida llamado `\myoutput` y abrimos el archivo de salida "output.txt" utilizando `\openout`. A continuación, utilizamos `\immediate\write` para escribir el mensaje "Message in the output file" en el archivo. Finalmente, cerramos el archivo de salida con `\closeout`.

Supongamos que queremos generar un comando especial para controlar la impresión de una imagen en el documento. Podemos hacerlo de la siguiente manera:

```
\special{pdf: image width=200pt height=150pt file=image.jpg}
```

En este ejemplo, utilizamos el comando `\special` para especificar una instrucción especial para el procesador de salida. En este caso, estamos indicando que se debe incluir una imagen llamada "image.jpg" con un ancho de 200 puntos y una altura de 150 puntos en el documento. El argumento de `\special` se trata como un elemento "whatsit".

Aplicaciones

Inspeccionar la entrada

TEX realiza un seguimiento del número de línea actual en el archivo de entrada actual utilizando el parámetro `\inputlineno` (en TEX3). Si el parámetro `\pausing` se establece en un valor positivo, TEX muestra cada línea que se ingresa en la pantalla de la terminal y permite al usuario insertar comandos. Estos comandos pueden incluir comandos `\show`, que proporcionan información sobre macros o variables. Los comandos insertados se tratan como si estuvieran directamente en el archivo fuente, por lo que no es necesario agregarles un prefijo ’i’ como sería necesario en mensajes de error.

```
\pausing=1
\show\myvariable
```

En este ejemplo, cuando se alcanza la línea con `\show\myvariable`, TEX mostrará la definición de la macro `\myvariable` en la pantalla de la terminal.

Comprobación de la existencia de archivos

Cuando se utiliza TEX para ingresar un archivo, es recomendable verificar si el archivo existe antes de intentar abrirlo. Esto se puede hacer utilizando la siguiente secuencia de comandos:

```
\newread\instream
\openin\instream=fname.tex
\ifeof\instream
  \message{File 'fname' does not exist!}
\else
  \closein\instream
  \input fname.tex
\fi
```

En este ejemplo, se abre un flujo de entrada con el nombre de archivo dado utilizando `\openin`. La prueba `\ifeof` se utiliza para verificar si se ha alcanzado el final del archivo en el flujo de entrada. Si la prueba se evalúa como verdadera, se muestra un mensaje que indica que el archivo no existe. De lo contrario, se cierra el flujo de entrada con `\closein` y se ingresa el archivo de manera segura utilizando `\input`.

Problemas de sincronización

La sincronización puede ser un problema crucial al tratar con operaciones de escritura y apertura/cierre de archivos en TEX. Esto es particularmente relevante al trabajar con archivos auxiliares, como los utilizados para referencias cruzadas en varios formatos de documentos.

Supongamos que necesitamos escribir en un archivo auxiliar durante la ejecución de TEX y luego ingresar el mismo archivo al final de la ejecución para diversos propósitos. Para asegurarnos de que todas las escrituras en el archivo se hayan realizado antes de cerrarlo, se puede utilizar la siguiente secuencia:

```
\par\vfil\penalty -10000
\immediate\closeout\auxfile
```

En esta secuencia, los primeros tres comandos activan el rutina de salida para cerrar la última página, asegurando que todas las escrituras se hayan realizado. Luego, se utiliza `\immediate\closeout` para cerrar inmediatamente el archivo auxiliar.

message versus immediate write16

TEX proporciona dos formas de mostrar mensajes al usuario: `\message` y `\immediate\write16`.

`\message` escribe el mensaje en la pantalla de la terminal. Los mensajes se concatenan y la línea se envuelve cuando se alcanza la longitud de línea (una constante de compilación de TEX). En la versión 2 de TEX, se pueden escribir un máximo de 1000 caracteres por mensaje.

`\immediate\write16` también escribe en la pantalla de la terminal, pero cada mensaje comienza en una nueva línea. El usuario puede forzar un salto de línea en el mensaje incluyendo el carácter con el código ASCII especificado por el parámetro `\newlinechar`. Este parámetro también funciona con `\message`.

```
\message{This is a message.}
\immediate\write16{This is another message.}
```

En este ejemplo, se mostrará el mensaje "This is a message." en la terminal, seguido de una nueva línea. El mensaje "This is another message." también se mostrará en una nueva línea.

Write dentro de una caja vertical

Cuando se realiza una operación de escritura al inicio de un \vtop, puede afectar la altura de la caja. Por ejemplo:

```
\vtop{\write\terminal{Hello!}\hbox{more text}}
```

En este ejemplo, el texto "Hello!" aparecerá en la pantalla de la terminal y el "more text" se posicionará debajo de la línea base.

Expansión y espacios en write y message

Tanto \write como \message expanden sus argumentos como si fueran el texto de reemplazo de un \edef. Esto significa que se expanden las macros y condicionales que aparecen en el argumento. Sin embargo, no se ejecutan comandos durante la expansión.

```
\def\a{b}
\message{\a}
```

En este ejemplo, el mensaje escrito será ’b’.

Las secuencias de control no expansibles (macros que no se pueden expandir más) se muestran con un espacio al final y se les agrega el carácter de escape \escapechar (por defecto, una barra invertida). Por ejemplo:

```
\message{\hbox\vbox!}
Esto escribirá '\hbox \vbox !'.
```

Las secuencias de control no definidas darán como resultado un error.

Las secuencias de control expansibles se pueden escribir con cierto cuidado. Aquí hay algunos ejemplos:

```
\message{\noexpand\ifx}
\message{\string\ifx}
{\let\ifx\relax \message{\ifx}}
```

Todos estos ejemplos escribirán ’\ifx’.

Tenga en cuenta que los espacios después de las secuencias de control expansibles se eliminan mediante el procesador de entrada, que ingresa al estado S después de encontrar una secuencia de control. Para insertar un espacio, puede definir una macro para ello y usarla en el mensaje o usar llaves:

```
\def\space{ } % en plain TeX
\message{\a\space\c}
\message{\a{ }\c}
```

Ambos ejemplos mostrarán ’b d’.

LaTeX AVANZADO

15.1 lplain.dtx

Definición de algunos constantes utilizando `\chardef`:

```
\chardef\@ne=1
\chardef\tw@=2
\chardef\thr@@=3
\chardef\sixt@@n=16
\chardef\@cclv=255
```

Definición de algunos constantes utilizando `\mathchardef`:

```
\mathchardef\@cclvi=256
\mathchardef\@m=1000
\mathchardef\@M=10000
\mathchardef\@MM=20000
```

15.2 Definición de otros macros (ltdefns.dtx)

Definiciones de comandos

El macro `\@gobble` se utiliza para descartar su argumento. También existen los macros para descartar dos, tres y cuatro argumentos Que se definen del siguiente modo:

```
\long\def \@gobble #1{}
\long\def \@gobbletwo #1#2{}
\long\def \@gobblethree #1#2#3{}
\long\def \@gobblefour #1#2#3#4{}
```

Definiciones internos de comandos

`\@xdblarg`, es definido de la siguiente forma:

```
\long\def\@xdblarg#1#2{#1[{#2}]{#2}}
```

Ejemplo 15.1

```
\makeatletter
\@xdblarg{a}{b}
\makeatother
```

a[b]b

`\@dblarg`, es un comando utilizado para verificar si el siguiente caracter es `[`. Su definición es de la forma:

```
\long\def\@dblarg#1{\kernel@ifnextchar[{#1}{\@xdblarg{#1}}}
```

Si el siguiente caracter es `[` entonces se ejecuta `{#1}`, y si no lo es, se ejecuta `\@xdblarg{#1}`.

Ejemplo 15.2

```
\makeatletter
\@dblarg{hola}[htbp]
\makeatother
```

hola[htbp]

Ejemplo 15.3

```
\makeatletter
\@dblarg{hola}asd
\makeatother
```

hola[a]asd

Ejemplo 15.4

```
\makeatletter
\@dblarg{hola}
\makeatother
```

hola[]

Ejemplo 15.5

```
\makeatletter
\@dblarg{hola}{arg}
\makeatother
```

hola[arg]arg

15.3 Contadores (ltalloc.dtx)

Esta sección trata con la asignación de contadores y otras variables.

El siguiente texto es de TEX plano:

1. `\z@`: Una dimensión o número cero. Es más eficiente escribir `\parindent\z@` que `\parindent 0pt`.
2. `\@ne`: El número 1.
3. `\m@ne`: El número -1.
4. `\tw@`: El número 2.
5. `\sixt@@n`: El número 16.
6. `\@m`: El número 1000.
7. `\@MM`: El número 20000.
8. `\@xxxii`: La constante 32.

15.4 Archivos (ltfiles.dtx)

`\document` : Lee los archivos .AUX y establece `\catcode` de @ en 12.

`\nofiles` : Suprime toda la salida de archivo estableciendo @filesw en false.

`\includeonly` : Hace que solo se lean las partes NAME1, ..., NAMEn por sus comandos `\include`. Funciona estableciendo partsw en true y estableciendo @partlist en NAME1, ..., NAMEn.

`\include` : Realiza una entrada de NAME a menos que @partsw sea true y NAME no esté en @partlist. Si @filesw es true, entonces dirige la salida de .AUX a NAME.AUX, incluyendo un punto de control al final.

`\input` : Es lo mismo que la entrada de TeX, excepto que permite llaves opcionales alrededor del nombre del archivo. En LaTeX2e, también evita el error primitivo "archivo no encontrado", si el archivo no se puede encontrar.

`\IfFileExists` : Si el archivo existe en el sistema, ejecuta then, de lo contrario, ejecuta else.

`\InputIfFileExists` : Si el archivo existe en el sistema, ejecuta then e ingresa NAME, de lo contrario, ejecuta else.

15.5 Contadores (ltcounts.dtx)

Estos son comandos para definir y usar contadores en LaTeX. Se definen los siguientes comandos:

`\newcounter` : para definir un nuevo contador.

`\setcounter` : para establecer el valor de un contador.

`\addtocounter` : para aumentar un contador en un número específico. Por ejemplo, `\addtocounter {contador}{2}` aumentaría el contador en 2.

`\stepcounter` : para aumentar un contador en 1.

`\refstepcounter` : para aumentar un contador en 1 y establecer su valor para su uso con `\label`.

`\value` : para acceder al valor numérico de un contador en TeX (en lugar de `\the{contador}`, que muestra la representación impresa del contador).

`\arabic` : para mostrar el valor del contador en números arábigos: 1, 2, 3, ...

`\roman` : para mostrar el valor del contador en números romanos: i, ii, iii, ...

`\Roman` : para mostrar el valor del contador en números romanos mayúsculas: I, II, III, ...

`\alph` : para mostrar el valor del contador en letras minúsculas: a, b, c, ...

`\Alph` : para mostrar el valor del contador en letras mayúsculas: A, B, C, ...

`\fnsymbol` : para mostrar el valor del contador como un símbolo: *, †, ‡, ...

`\counterwithin` : para restablecer un contador cada vez que otro contador es aumentado. Por ejemplo, `\counterwithin{contador}{contador_padre}` reseteará el contador cada vez que se aumente el `contador_padre`.

`\counterwithout` : para quitar un contador de la lista de reseteo de otro contador.

Estos comandos permiten controlar y personalizar la forma en que se muestran los contadores en un documento LaTeX.

15.6 Longitudes (ltlength.dtx)

`\newlength` : Este comando permite declarar una nueva longitud, es decir, una nueva medida de distancia. Se usa de la siguiente manera:

```
\newlength{\nombrelongitud}
```

`\setlength` : Este comando permite establecer el valor de una longitud. Se usa de la siguiente manera:

```
\setlength{\nombrelongitud}{valor}
```

`\addtolength` : Este comando permite aumentar el valor de una longitud existente. Se usa de la siguiente manera:

```
\addtolength{\nombrelongitud}{valor}
```

`\settowidth` Este comando permite establecer la longitud a la anchura de un caja que contiene un objeto determinado. Se usa de la siguiente manera:

```
\settowidth{\nombrelongitud}{objeto}
```

`\settoheight` : Este comando permite establecer la longitud a la altura de una caja que contiene un objeto determinado. Se usa de la siguiente manera:

```
\settoheight{\nombrelongitud}{objeto}
```

`\settodepth` : Este comando permite establecer la longitud a la profundidad de una caja que contiene un objeto determinado. Se usa de la siguiente manera:

```
\settodepth{\nombrelongitud}{objeto}
```

15.7 fntcmd.dtx

en el archivo "ltfntcmd" son parte del código del kernel de LATEX2e/NFSS2. Además, está diseñado para ser una documentación para los escritores de paquetes, ya que muestra cómo definir comandos de cambio de fuente de alto nivel usando un pequeño número de funciones de creación.

En resumen, los comandos en el archivo "ltfntcmd" son parte integral del sistema LATEX y están destinados a ser una guía para los desarrolladores de paquetes al mostrar cómo implementar cambios de fuentes en sus proyectos.

Introducción

En esta sección se estudia como funcionan los cambios de fuente en el sistema TEX, y cómo estos han sido implementados en LATEX3. Los cambios de fuente, como `\bfseries`, `\sffamily`, etc., son declaraciones, lo que significa que su alcance está delimitado por la estructura de agrupación, ya sea por el siguiente `\end` de algún entorno o explícitamente usando un grupo, por ejemplo, escribiendo algo como `{\bfseries...}` en el código fuente. Si se comete el error de escribir `\bfseries{...}` (pensando en `\bfseries` como un comando con un argumento), el resultado es sorprendente.

Las declaraciones de fuente son un artefacto del sistema TEX y por varias razones es mejor evitarlas a nivel de usuario siempre que sea posible. En LATEX3, probablemente serán reemplazadas todas por entornos y por comandos de fuente con un argumento.

Este archivo define una función de creación para estos cambios de fuente declarativos. Esta función crea comandos que se pueden usar tanto en matemáticas como en texto. Este archivo también define una serie de comandos de alto nivel (todos comenzando con `\text..`) que tienen un argumento y tipografían este argumento de la forma solicitada. Por lo tanto, estos comandos son para tipografiar piezas cortas de texto en una familia, serie o forma específicas. Todos estos se producen como ejemplos del uso de una función de creación, que también está definida en este archivo.

Por lo tanto, al usar estos comandos, no se tiene que preocupar por olvidar la corrección itálica al cambiar de fuente. Solo en muy pocas situaciones esta espacio adicional es incorrecto, pero por ejemplo, la mayoría de los tipógrafos recomiendan omitir la corrección itálica si un pequeño signo de puntuación, como una coma, sigue directamente el cambio de fuente. Dado que la cantidad de corrección necesaria es en parte cuestión de gusto, se puede definir en qué situaciones se debe suprimir la corrección itálica. Esto se hace colocando los caracteres que deben cancelar una corrección itálica anterior en la lista `\nocorrlist`. La definición predeterminada para esta lista es producida por lo siguiente.

```
\newcommand \nocorrlist {,.}
```

Una posible definición para \nocorrlist podría ser:

```
\def\nocorrlist{.,;?!}
```

Esto significa que los caracteres ".", ";", "?" y "!" cancelarán cualquier corrección itálica precediendo. Por lo tanto, si se utiliza un comando de alto nivel para cambiar la fuente, seguido de uno de estos caracteres, no habrá corrección itálica aplicada.

Inicialización

La definición de \normalsize es para producir un error. La línea 143 define \normalsize como una función que muestra un mensaje de error indicando que la función de tamaño de fuente \normalsize no está definida y que probablemente haya algo mal con el archivo de clase. Esto significa que \normalsize no está disponible para su uso y producirá un error si se intenta utilizar.

Entornos miscelaneos

Entornos

La lectura de datos de archivos auxiliares (como .toc) normalmente ocurre en modo vertical y por lo tanto no importa si los finales de línea son convertidos a espacios por TEX durante ese proceso.

Sin embargo, especialmente el archivo .toc puede ser leído en modo L-R (en casos en los que el comando \tableofcontents intenta colocar, por ejemplo, una lista de subsecciones como un párrafo). En ese caso, los saltos de línea después de una línea como

```
\contentsline {subsubsection}{\numberline{1.1.1}A C-head}{2}
```

Esta línea puede resultar en espacios spurious (por ejemplo, cuando ese nivel no está incluido). Eso podría solucionarse leyendo el archivo usando \endlinechar=-1, pero eso tiene el peligro de descartar algunas líneas finales válidas que deben convertirse en espacios (por ejemplo, cuando el usuario editó el TOC y luego usó \nofiles para preservarlo).

Por lo tanto, la aproximación tomada es la siguiente:

1. \addcontentsline agrega el comando \protected@file@percent al final del segundo argumento de \@writefile que se escribe en el archivo .aux. Como indica su nombre, se trata de una macro protegida, por lo que no cambia si se escribe.

2. Cuando el archivo .aux se lee de nuevo al final de la ejecución, se ejecuta \@writefile y escribe su segundo argumento sin modificar en el archivo con la extensión dada por su primer argumento. O eso era cómo era en el pasado.

3. En cambio, cambiamos ligeramente \@writefile: básicamente, mira el segundo argumento y si el último token en ese argumento es \protected@file@percent, entonces se reemplaza por un

carácter porcentual y eso es lo que se escribirá. Si no (por ejemplo, si los datos provienen de un `\addtocontents` emitido por el usuario o de algún paquete que utiliza `\@writefile` para escribir sus propios archivos), el comando se comporta exactamente como antes.

15.8 Listas (ltlist.dtx)

Listas y entornos relacionados

Los comandos genéricos para crear un entorno con sangría, como enumerate, itemize, quote, etc., son:

```
\list{⟨LABEL⟩}{⟨COMMANDS⟩} ... \endlist
```

El usuario invoca este comando como el entorno "list". El argumento `<ETIQUETA>` especifica el tipo de etiquetado de los elementos de la lista, como números o viñetas. El argumento `<COMANDOS>` contiene comandos para cambiar los parámetros de espaciado horizontal y vertical.

Cada elemento del entorno se comienza con el comando

```
\item[ITEMLABEL]
```

que produce un elemento etiquetado por ITEMLABEL. Si el argumento falta, entonces se usa el argumento LABEL de la comando `\list` como etiqueta del elemento.

La etiqueta se forma poniendo `\makelabel{<ITEMLABEL>}` en un hbox cuya anchura es su anchura natural o `\labelwidth`, lo que sea más grande. El comando `\list` define `\makelabel` con la definición predeterminada:

```
\makelabel{⟨ARG⟩} == BEGIN \hfil ARG END
```

lo que, para una etiqueta con una anchura menor que `\labelwidth`, coloca la etiqueta justificada a la derecha, `\labelsep` a la izquierda del texto del elemento. Sin embargo, `\makelabel` puede ser `\let` a otro comando mediante el argumento COMMANDS de `\list`.

El comando `\usecounter{foo}` inicializa el contador "foo" a cero y lo incrementa con cada comando `\item` sin argumento. Las etiquetas se pueden hacer referencia en la lista con el comando `\label`.

Cuando sales de un entorno de lista o `list` en LaTeX, ya sea volviendo a una lista de nivel superior o al modo de texto normal, LaTeX comienza un nuevo párrafo si y solo si dejas una línea en blanco después del comando `\end`. Esto se logra mediante el comando `\@endparenv`. Las líneas en blanco se ignoran en cualquier otro lugar razonable, es decir:

- Entre `\begin{list}` y el primer `\item`.
- Entre el `\item` y el texto de ese ítem.
- Entre el final del último ítem y `\end{list}`

Para un entorno como “quotation”, en el que los ítems no tienen etiquetas, todo el entorno es un solo ítem. Se define del siguiente modo:

```
\quotation == \list{}{...}\item\relax
```

(Observa el `\relax`, en caso de que el primer carácter en el entorno sea ’[’). Los parámetros de espaciado brindan mucha flexibilidad en el diseño del formato, incluida la capacidad de permitir que la sangría del primer párrafo sea diferente de la de los siguientes.

El entorno “trivlist” es equivalente a un entorno de lista cuyo segundo argumento establece los siguientes valores de parámetros:

1. `\leftmargin = 0`: no causa sangría del margen izquierdo
2. `\labelwidth = 0`
3. `\itemindent = 0`, junto con una etiqueta nula, hace que el primer párrafo no tenga sangría. Los párrafos siguientes tienen sangría de `\parindent`. Para dar al primer párrafo la misma sangría, establece `\itemindent = \parindent` antes del `\item[]`.

Cada `\item` en un entorno trivlist debe tener un argumento, y en muchos casos, este será el argumento nulo (`\item[]`). El entorno trivlist se utiliza principalmente para entornos de párrafos, como verbatim, en los que no hay cambio de margen. Proporciona el mismo espaciado vertical que el entorno de lista y funciona razonablemente bien cuando ocurre inmediatamente después de un comando `\item` en una lista de nivel superior.

List y trivlist

Los siguientes variables se utilizan dentro de un entorno lista:

- `\@totallleftmargin`: La distancia que la margen izquierda predominante está sangrada desde la margen izquierda exterior.

  Ejemplo de uso: Supongamos que queremos establecer un sangrado adicional de 1 cm en la lista. Podemos hacerlo de la siguiente manera:

  ```
  \setlength{\@totalleftmargin}{\dimexpr\@totalleftmargin + 1cm}
  ```

- `\linewidth`: El ancho de la línea actual. Debe inicializarse en `\hsize`.

  Ejemplo de uso: Si deseamos ajustar el ancho de la línea a la mitad del ancho del texto, podemos hacer lo siguiente:

  ```
  \setlength{\linewidth}{0.5\textwidth}
  ```

- `\@listdepth`: Un contador que mantiene el nivel de anidamiento actual de la lista.

  Ejemplo de uso: Podemos imprimir el valor actual de @listdepth de la siguiente manera:

```
El nivel de anidamiento de la lista es \@listdepth.
```

- `\makelabel`: Una macro con un solo argumento, utilizada para generar la etiqueta a partir del argumento (implícito o explícito) del comando `\item`. Se inicializa en `\@mklab` por el comando `\list`. Este comando debe producir algún estiramiento, es decir, un `\hfil`.

  Ejemplo de uso: Podemos personalizar el formato de la etiqueta utilizando `\makelabel`. Por ejemplo, si queremos que las etiquetas estén en negrita, podemos hacer lo siguiente:

  ```
  \renewcommand{\makelabel}[1]{\textbf{#1}}
  ```

- `\@inlabel`: Un interruptor que es falso excepto entre el momento en que se encuentra un `\item` y el momento en que TEX entra realmente en modo horizontal. Debe ser probado por comandos que puedan ser afectados por el uso de `\everypar` del entorno de lista.

  Ejemplo de uso:

  ```
  \if@inlabel
     Estamos en el proceso de etiquetado de un elemento de lista.
  \else
     No estamos en el proceso de etiquetado.
  \fi
  ```

- `\box@labels`: Cuando `\@inlabel = true`, contiene las etiquetas que se deben poner en `\everypar`.

- `@noparitem`: Un interruptor establecido por `\list` cuando `\@inlabel = true`. Maneja el caso de que una lista sea lo primero en un ítem.

- `@noparlist`: Un interruptor establecido en true para una lista que comienza un ítem. No se agrega espacio `\topsep` antes ni después de los `\item` de dicha lista.

- `@newlist`: Establecido en true por `\list`, establecido en falso por el primer texto (por `\everypar`).

- `@noitemarg`: Se establece en true al ejecutar un `\item` sin argumento explícito. Se utiliza para ahorrar espacio. Para ahorrar tiempo, se pueden hacer dos comandos `@item` separados.

- `@nmbrlist`: Establecido en true por el comando `\usecounter`, hace que la lista tenga numeración.

- `@listctr`: Se define por `\usecounter` con el nombre del contador.

- `@noskipsec`: Un interruptor establecido en true por un comando de sección cuando está creando un encabezado en el texto con `\everypar`.

A lo largo de un entorno de lista, `\hsize` es el ancho de la línea actual, medido desde la margen izquierda exterior hasta la margen derecha exterior. Los entornos como tabbing deben usar `\linewidth` en lugar de `\hsize`.

Aquí están los parámetros de una lista que se pueden establecer mediante comandos en el argumento COMMANDS de `\list`. Estos parámetros son saltos o dimensiones TeX (definidos por `\newskip` o

\newdimen), por lo que se pueden usar los comandos TEX o LATEX habituales para establecerlos. Los comandos se ejecutarán en vmode solo si el \list fue precedido por un \par (o algo similar a un \end{list}), por lo que los parámetros de espaciado se pueden establecer según si la lista está dentro de un párrafo o es su propio párrafo.

15.9 Comandos de seccionamiento (ltsect.dtx)

Título

\title es un comando de LaTeX que se utiliza para declarar el título de un documento.

\author es un comando de LaTeX que se utiliza para declarar el autor de un documento.

\date es un comando de LaTeX que se utiliza para declarar la fecha en la que se escribió un documento.

\thanks es un comando de LaTeX que se utiliza para agregar una nota al pie de página, por ejemplo, para agradecimientos o notificaciones de direcciones.

\and es un comando de LaTeX que se utiliza para separar varios autores en un documento.

\maketitle es un comando de LaTeX que se utiliza para producir el título real del documento, utilizando la información previamente guardada con los otros comandos.

Secciones

El comando

```
\@startsection{⟨name⟩}{⟨level⟩}{⟨indent⟩}
{⟨beforeskip⟩}{⟨afterskip⟩}
{⟨style⟩}*[⟨altheading⟩]{⟨heading⟩}
```

es la madre de todos los comandos de sección de nivel de usuario en LaTeX. La parte después del asterisco, incluido el asterisco, es opcional.

Este comando es una herramienta poderosa para crear secciones y subtítulos en un documento de LaTeX. Cada parámetro en el comando controla un aspecto diferente de la sección, como el nombre, el nivel de profundidad, el sangrado, el espacio antes y después de la sección, el estilo, el título alternativo y el título principal. La parte opcional después del asterisco permite especificar un título alternativo para la sección.

En resumen, el comando \@startsection es una herramienta esencial para la creación de secciones y subtítulos en documentos de LaTeX.

```
\secdef{⟨unstarcmds⟩}{⟨unstarcmds⟩}{⟨starcmds⟩}
```

\secdef

La macro \secdef se utiliza para definir \chapter y \section sin utilizar el comando \@startsection.

```
\def\chapter{...\secdef\<starcmd>\<unstarcmd>}
```

```
2 \def\<starcmd>[#1]#2{...}
3 \def\<unstarcmd>#1{...}
```

La instrucción

```
\@hangfrom{⟨text⟩}
```

pone ⟨text⟩ en una caja y crea una sangría colgante para el material que sigue hasta el primer `\par`. Se debe usar en modo vertical.

Tabla de contenido

Convenciones

`\tf@<foo>` = número de archivo para la salida de la tabla foo. El archivo solo se abre si `@filesw` = `true`.

Comandos

Se requiere definir una macro `\l@<type>{<entry>}{<page>}` por el estilo del documento para hacer una entrada de tipo <type> en una tabla de contenidos, etc. Por ejemplo, el estilo del documento debería definir `\l@chapter`, `\l@section`, etc.

Nota: Cuando se utiliza el comando `\protect` en la ⟨entry⟩ o ⟨text⟩ de uno de los comandos a continuación, hace que la secuencia de control siguiente se escriba en el archivo sin ser expandida. La secuencia se expandirá cuando se procese la entrada de la tabla de contenidos.

Sorpresa: Dentro de un comando `\addcontentsline` o `\addtocontents`, los comandos: `\index`, `\glossary` y `\label` no son operativos. Esto podría causar un problema si el usuario pone un `\index` o `\label` en uno de los comandos que escriba o en el argumento opcional "versión corta" de un comando `\section` o `\caption`.

El comando

`\@starttoc`

```
\@starttoc{⟨ext⟩}
```

se utiliza para definir los comandos: `\tableofcontents`, `\listoffigures`, etc.

Por ejemplo: `\@starttoc{lof}` se utiliza en `\listoffigures`. Este comando lee el archivo .⟨ext⟩ y establece para escribir el nuevo archivo .⟨ext⟩.

El comando

`\addcontentsline`

```
\addcontentsline{⟨table⟩}{⟨type⟩}{⟨entry⟩}
```

permite al usuario agregar su propia entrada a una tabla de contenidos, etc. El comando agrega la entrada

```
\contentsline{⟨type⟩}{⟨entry⟩}{⟨page⟩}{}
```

al archivo .<table>.

Esta macro está implementada como una aplicación de \addtocontents. Tenga en cuenta que \thepage no es expandible durante \protected@write, por lo que se obtiene el número de página en el momento del \shipout.

El comando

```
\@dottedtocline{⟨level⟩}{⟨indent⟩}{⟨numwidth⟩ }{⟨title⟩}{⟨page⟩}
```

produce una línea de tabla de contenido con los siguientes parámetros:

- Nivel <level>: Si el nivel <level> es mayor que la profundidad máxima de la tabla de contenidos (\c@tocdepth), entonces no se produce ninguna línea.
- Sangría <indent>: Indentación total desde el margen izquierdo.
- Ancho del número <numwidth>: Ancho del recuadro para el número si el título <title> tiene un comando \numberline. A partir del 25 de enero de 1988, esto también representa la cantidad de sangría adicional agregada a las segundas y siguientes líneas de una entrada de varias líneas.
- Título <title>: Contenido de la entrada.
- Número de página <page>: Número de página.

Utiliza los siguientes parámetros, los cuales deben ser establecidos por el estilo del documento. Deben ser definidos con comandos \def.

- Ancho del número de página <pnumwidth>: Ancho del recuadro en el que se coloca el número de página.
- Sangría del margen derecho <tocrmarg>: Sangría del margen derecho para todas las líneas excepto la última de las entradas de varias líneas.
- Separación entre puntos <dotsep>: Separación entre los puntos, en unidades mu. Debe ser definido con un número, por ejemplo, 2 o 1.7.

15.10 Estilos de páginas (ltpage.dtx)

Comandos de estilo de página

\pagestyle{<style>}: Establece el estilo de página de las páginas actuales y las siguientes.

\thispagestyle{<estilo>}: Establece el estilo de página de la página actual solo en el estilo.

Para definir un estilo de página estilo, debe definir \ps@<style> para establecer los parámetros de estilo de página.

Cómo un estilo de pagina crea encabezados y pies de página

El comando `\ps@...` define las macros `\@oddhead`, `\@oddfoot`, `\@evenhead` y `\@evenfoot` para definir los encabezados y pies de página.

Para hacer que los encabezados se determinen por los comandos de seccionamiento, el estilo de página define los comandos `\chaptermark`, `\sectionmark`, etc., donde `\chaptermark{<text>}` es llamado por `\chapter` para establecer una marca. Los comandos `\...mark` y `\...head` se definen con la ayuda de las siguientes macros.

Convenciones de marcado

LATEX extiende la funcionalidad `\mark` de TEX produciendo dos tipos de marcas, una "izquierda" y una "derecha", utilizando los siguientes comandos:

1. `\markboth{<left>}{<right>}` agrega ambas marcas.
2. `\markright{<right>}` agrega una marca "derecha".
3. `\leftmark` se usa en la rutina de salida, obtiene la marca "izquierda" actual. Funciona como `\botmark` de TEX.
4. `\rightmark` se usa en la rutina de salida, obtiene la marca "derecha" actual. Funciona como `\firstmark` de TEX.

Los comandos de marcado funcionan razonablemente bien para marcas derechas "numeradas dentro" de marcas "izquierdas" - por ejemplo, la marca "izquierda" cambia con un comando `\chapter` y la marca "derecha" cambia con un comando `\section`. Sin embargo, produce resultados algo anómalos si ocurren 2 `\markboth` en la misma página.

Comandos como `\tableofcontents` que deben establecer las marcas en algunos estilos de página usan un comando `\@mkboth`, que es `\let` por el comando de estilo de página (`\ps@...`) a `\markboth` para establecer el encabezado o a `\@gobbletwo` para no hacer nada.

Esta frase se refiere a cómo LATEX extiende la funcionalidad de marcado de TEX, permitiendo la producción de dos tipos de marcas: izquierda y derecha. Los comandos `\markboth`, `\markright`, `\leftmark` y `\rightmark` se utilizan para establecer y obtener las marcas. Sin embargo, si dos `\markboth` ocurren en la misma página, los resultados pueden ser anómalos.

El comando `\@mkboth` se utiliza en algunos comandos como `\tableofcontents` para establecer las marcas en algunos estilos de página, y es `\let` por el comando de estilo de página (`\ps@...`) a `\markboth` para establecer el encabezado o a `\@gobbletwo` para no hacer nada.

Ejemplos aplicativos

Ejemplo 15.6

```
\documentclass{book}
\usepackage[margin=2cm, paperwidth=14cm, paperheight=18cm]{geometry}
\usepackage{lipsum}
\begin{document}
\markboth{Izquierda(left)}{Derecha(right)}
\lipsum[1]

\leftmark

\rightmark

\firstmark

\newpage
\leftmark

\rightmark

\firstmark
\end{document}
```

Derecha(right) 1

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

2 *Izquierda(left)*

Izquierda(left)
Derecha(right)
Izquierda(left)Derecha(right)

Ejemplo 15.7

```
\documentclass{book}
\usepackage[margin=2cm, paperwidth=14cm, paperheight=18cm]{geometry}
\usepackage{lipsum}
\makeatletter
\def\ps@mystyle{%
  \def\@oddfoot{Pie de página en páginas impares}
  \def\@evenfoot{Pie de página en páginas pares}
  \def\@evenhead{Encabezado en páginas pares}
  \def\@oddhead{Encabezado en páginas impares}
}
\makeatother
\begin{document}
\pagestyle{mystyle}
%forma alternativa
%\makeatletter
%\ps@mystyle
%\makeatother
\lipsum[1-5]
\end{document}
```

Encabezado en páginas impares

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar

Pie de página en páginas impares

Encabezado en páginas pares

lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula.

Fusce mauris. Vestibulum luctus nibh at lectus. Sed bibendum, nulla a faucibus semper, leo velit ultricies tellus, ac venenatis arcu wisi vel nisl. Vestibulum diam. Aliquam pellentesque, augue quis sagittis posuere, turpis lacus congue quam, in hendrerit risus eros eget felis. Maecenas eget erat in sapien mattis porttitor. Vestibulum porttitor. Nulla facilisi. Sed a turpis eu lacus commodo facilisis. Morbi fringilla, wisi in dignissim interdum, justo lectus sagittis dui, et vehicula libero dui cursus dui. Mauris tempor ligula sed lacus. Duis cursus enim ut augue. Cras ac magna. Cras nulla. Nulla egestas. Curabitur a leo. Quisque egestas wisi eget nunc. Nam feugiat lacus vel est. Curabitur consectetuer.

Pie de página en páginas pares

15.11 Clases (ltclass.dtx)

Introducción

Este fragmento de código describe un archivo que forma parte de un sistema de formato de texto llamado LATEX2e. Su función principal es reemplazar la antigua declaración `\documentstyle` para mejorar la compatibilidad con documentos antiguos. Además, se mencionan dos tipos de archivos de estilo en LATEX2e: los "archivos de clase" y los "paquetes", donde los archivos de clase definen elementos y proporcionan un formato predeterminado, y los paquetes proporcionan funcionalidad adicional. También se menciona que los paquetes en LATEX2e pueden tener opciones, pero su implementación puede ser compleja.

Interfaz de usuario

```
\documentclass[⟨main-option-list⟩]{⟨class⟩}[⟨version⟩]
\documentstyle[⟨main-option-list⟩]{⟨class⟩}[⟨version⟩]
\usepackage[⟨package-option-list⟩]{⟨package-list⟩}[⟨version⟩]
```

Procesamiento de opciones

Primero se procesan las opciones locales y globales que se han declarado (usando \DeclareOption como se describe a continuación).

1 En el caso de \ProcessOptions, se procesan en el orden en que se declararon en la clase o paquete.

En el caso de \ProcessOptions*, se procesan en el orden en que aparecen en las listas de opciones. Primero las opciones globales y luego las locales.

2 Cualquier opción local restante se trata usando la opción predeterminada (declarada usando la declaración \DeclareOption* descrita a continuación).

Para las clases de documentos, esto normalmente no hace nada, pero registra la opción en una lista de opciones no utilizadas.

Para los paquetes, esto suele producir un error.

Finalmente, cuando se alcanza \begin{document}, si hay alguna opción global que no ha sido utilizada por la clase o por algún paquete, el sistema producirá una advertencia. Es decir, si hay alguna opción que se ha especificado en el preámbulo del documento pero que no ha sido procesada por ninguna clase o paquete, el sistema le informará al usuario. Esto es para que el usuario sepa que alguna opción que esperaba utilizar no se está utilizando y pueda corregir cualquier problema o error.

Interfaces de clases y paquetes

1 \ProvidesClass{⟨name⟩}[⟨version⟩]

2 \ProvidesPackage{<name>}[<version>]

3 \RequirePackage[<options>]{<name>}[<version>]

4 \LoadClass

5 \PassOptionsToPackage{⟨options⟩}{⟨package⟩}

6 \PassOptionsToClass

7 \LoadClassWithOptions{⟨name⟩}{⟨version⟩}

8 \RequirePackageWithOptions

9 \IfPackageLoadedTF{⟨package⟩}{⟨true⟩}{⟨false⟩}

10 \IfClassLoadedTF{⟨class⟩}{⟨true⟩}{⟨false⟩}

11 \@ifpackageloaded

12 \@iffclassloaded

13. `\IfPackageAtLeastTF{⟨package⟩}{⟨date⟩}{⟨true⟩}{⟨false⟩}`
14. `\IfClassAtLeastTF{⟨class⟩}{⟨date⟩}{⟨true⟩}{⟨false⟩}`
15. `\@ifpackagelater`
16. `\@ifclasslater`
17. `\IfFormatAtLeastTF{⟨date⟩}{⟨true⟩}{⟨false⟩}`
18. `\IfPackageLoadedWithOptionsTF{⟨package⟩}{⟨true⟩}{⟨false⟩}`
19. `\IfClassLoadedWithOptionsTF{⟨class⟩}{⟨true⟩}{⟨false⟩}`
20. `\@ifpackagewith`
21. `\@ifclasswith`

Entrada de macros seguros

1. `\InputIfFileExists{⟨file⟩}{⟨then⟩}{⟨else⟩}`: Entra en ⟨file⟩ si existe. Inmediatamente antes de la entrada, se ejecuta ⟨then⟩. De lo contrario, se ejecuta ⟨else⟩.
2. `\IfFileExists`: Igual que arriba, pero no entra al archivo. Una cosa que puede interesarte poner en la cláusula ⟨else⟩ es `\@missingfileerror`.
3. `\input`: Esto ha sido redefinido desde la definición de LATEX2.09, en términos de los nuevos comandos `\InputIfFileExists` y `\@missingfileerror`.
4. `\listfiles`: Dar esta declaración en la preámbulo hace que se incluya una lista de todos los archivos entrados a través de los comandos de "entrada segura". Cualquier cadena especificada en el argumento opcional a `\ProvidesPackage` se incluirá junto con el nombre del archivo. De esta manera, los archivos en distribuciones estándar (y otras no estándar) pueden incluir cadenas informativas en este argumento.

15.12 ltshipout

En esta sección nos enfocaremos en estudiar el envío de página final a los archivos .dvi y .pdf que se basa sobre las ideas de Heiko Oberdiek en su paquete atbegshi y atenddvi aunque el interfaz de estos paquetes y del nucleo de LaTeX son algo diferentes.

Sobrecarga de shipout

`\shipout`

La primitiva `\shipout` en LaTeX a diferencia del sistema TeX es un poco más complicado. Esta primitiva envía una caja independientemente de cómo se haya construido. Esta caja que se está pretendiendo enviar al archivo de salida se puede manipular atravez de hooks.

Cada vez que se envía una página a través de la primitiva shipout se registra y el número total de envíos está disponible en una variable y un contador de LaTeX y estas variables son solo de lectura.

15.13 Output routine (ltoutput.dtx)

Parámetros de una página

"Page Layout Parameters" son parámetros que se utilizan para controlar el diseño de la página en un documento. Aquí está una explicación detallada de cada parámetro, aunque algunos de estos parámetros ya conocemos:

`\topmargin` : es un espacio extra agregado a la parte superior de la página.

`@twoside` : es una variable booleana, T si la impresión es a dos caras.

`\oddsidemargin` : si @twoside = T, entonces se agrega un espacio extra a la izquierda de las páginas impares. En caso contrario, se agrega a la izquierda de todas las páginas.

`\evensidemargin` : si @twoside = T, entonces se agrega un espacio extra a la izquierda de las páginas pares.

`\headheight` : es la altura de la cabecera o encabezado.

`\headsep` : es la separación entre la cabecera y el texto.

`\footskip` : es la distancia de separación entre la línea base del último texto y la línea base del pie de página. Nota la diferencia entre `\footskip` y `\headsep`.

`\textheight` : es la altura del texto en la página, excluyendo la cabecera y el pie de página.

`\textwidth` : es la anchura de la impresión en la página.

`\columnsep` : si @twocolumn = T, entonces es la anchura del espacio entre columnas.

`\columnseprule` : si @twocolumn = T, entonces es la anchura de la regla entre columnas (0 si no hay ninguna).

`\columnwidth` : si @twocolumn = T, entonces es (`\textwidth` - `\columnsep`)/2. En caso contrario, es `\textwidth`. Se establece mediante los comandos `\twocolumn` y `\onecolumn`.

Las dimensiones dimensiones `\paperwidth` y `\paperheight` son utilizado para crear un nuevo tipo de papel.

Parámetros de estilos de página

- `\floatsep`: Espacio dejado entre flotantes. Este parámetro establece la distancia vertical mínima entre flotantes en una misma página.

- `\textfloatsep`: Espacio entre el último flotante en la parte superior o el primer flotante en la parte inferior y el texto. Este parámetro determina la distancia vertical entre los flotantes y el texto que los rodea.

- `\topfigrule`: Comando para colocar una línea (o lo que sea) entre flotantes en la parte superior de la página y el texto. Se ejecuta en modo vertical interno justo antes del salto `\textfloatsep` que separa los flotantes del texto. Debe ocupar cero espacio vertical. (Ver `\footnoterule`).

- `\botfigrule`: Similar a `\topfigrule`, pero se coloca después del salto `\textfloatsep` que separa el texto de los flotantes en la parte inferior de la página.

- `\intextsep`: Espacio dejado en la parte superior e inferior de un flotante dentro del texto. Este parámetro determina la distancia vertical mínima entre un flotante dentro del texto y el texto que lo rodea.

- `\dblfloatsep`: Espacio entre flotantes de doble columna. Este parámetro establece la distancia vertical mínima entre flotantes de doble columna en una misma página.

- `\dbltextfloatsep`: Espacio entre los flotantes de doble columna en la parte superior y el texto. Este parámetro determina la distancia vertical entre los flotantes de doble columna y el texto que los rodea.

- `\dblfigrule`: Similar a `\topfigrule`, pero para flotantes de doble columna.

- `\@fptop`: Pegamento para colocar en la parte superior de un flotante en una columna. Debe ser 0pt + stretch.

- `\@fpsep`: Pegamento para colocar entre flotantes en una columna.

- `\@fpbot`: Pegamento para colocar en la parte inferior de un flotante en una columna.

- `\@dblfptop`, `\@dblfpsep`, `\@dblfpbot`: Análogos a `\@fptop`, `\@fpsep` y `\@fpbot` para flotantes de doble columna.

Footnotes

Para insertar una pie de página se utiliza al macro `\insert\footins`.

Ejemplo 15.8

```
\documentclass{article}
\usepackage[paperwidth=10cm,paperheight=2cm,margin=2mm]{geometry}
\begin{document}
Este es un ejemplo de texto con una nota al pie. \insert\footins{Esta es la nota al pie de página.}
\end{document}
```

Este es un ejemplo de texto con una nota al pie.

Esta es la nota al pie de página.

Page layout switches and macros

El booleano `@twocolumn` determina si el documento se debe mostrar en formato de dos columnas por página o no.

Page style macros and switches

`\@oddhead`:

```
IF @twoside = T
  THEN genera el encabezado de las páginas impares.
  ELSE genera el encabezado de todas las páginas.
```

`\@evenhead`:

```
IF @twoside = T
  THEN genera el encabezado de las páginas pares.
```

`\@oddfoot`:

```
IF @twoside = T
  THEN genera el pie de página de las páginas impares.
  ELSE genera el pie de página de todas las páginas.
```

`\@evenfoot`:

```
IF @twoside = T
  THEN genera el pie de página de las páginas pares.
```

`\@specialpage`: Si su valor es foo, entonces

```
IF @specialpage = T
  THEN se ejecuta el comando \mintinline{tex}{\ps@foo} para restablecer
   ↪ temporalmente los parámetros del estilo de página antes de componer la
   ↪ página actual.
```

Float placement parameters

La macro \@floatplacement se encarga de establecer los parámetros relacionados con la colocación de objetos flotantes, como tablas y figuras. Uno de los parámetros que se establece es \@colht, que representa la altura de la página o columna en construcción. Valor de \@colht:

- Para una página de una sola columna, es igual a \textheight.
- Para una página de dos columnas, es igual a \textheight - altura de los flotantes de doble columna en la página.

Otros parámetros que se establecen son:

- \@topnum := G Número máximo de flotantes permitidos en la parte superior de una columna.
- \@toproom := G Máximo espacio de la parte superior de una columna dedicado a los flotantes (se excluye la separación \textfloatsep debajo de los flotantes y la separación \floatsep entre ellos). Para la salida de dos columnas, debe calcularse como una función de \@colht.
- \@botnum, \@botroom: Análogos a los anteriores, solo que ahora trata de la parte inferior de la página.
- \@colnum := G Número máximo de flotantes permitidos en una columna.
- \@textmin := L Cantidad mínima de texto (excluyendo las notas al pie) que debe aparecer en una página de texto.
- \@fpmin := L Altura mínima de los flotantes en una columna de flotantes.

La macro \@dblfloatplacement establece los siguientes parámetros:

- \@dbltopnum := G Número máximo de flotantes de doble columna permitidos en la parte superior de una página de dos columnas.
- \@dbltoproom := G Altura máxima de los flotantes de doble columna permitidos en la parte superior de una página de dos columnas.
- \@fpmin := L Altura mínima de los flotantes en una columna de flotantes.

También es posible realizar las siguientes asignaciones locales cuando sea necesario:

```
\@fptop := \@dblfptop
\@fpsep := \@dblfpsep
\@fpbot := \@dblfpbot
```

Variables de rutina de salida(output routine variables)

- \@colht. La altura total de la columna actual. Es igual a \textheight cuando es de una sola columna. Es igual a \textheight menos la altura de los flotantes cuando es de dos columnas. Esta inicializado a \textheight.
- \@colroom. La altura disponible en la columna actual para el texto y las notas al pie. Es igual a \@colht menos la altura de todos los flotantes ubicados en la parte superior e inferior de la columna actual.
- \@textfloatsheight. La altura total de los flotantes en el texto en la página actual.
- \footins. Número de inserción de notas al pie.
- \@maxdepth. Valor guardado de \maxdepth de TeX.

Calling the output routine

El otr es llamado por ruptura de página de TeX o por una macro que coloca una penalización en la lista de salida.

Tabla 15.29

Penalty	reason
-10000	\pagebreak \newpage
-10001	\clearpage (\penalty -10000 \vbox{} \penalty -10001)
-10002	Inserción de flotante, llamado desde modo horizontal
-10003	Inserción de flotante, llamado desde modo vertical
-10004	Inserción de flotante

Nota: Un float o marginpar pone la siguiente secuencia en la lista de salida:

- una penalización de -10004,
- una caja nula \vbox,
- una penalización de -10002 o -10003.

Esto resuelve dos problemas especiales: Si el float viene justo después de un \newpage o \clearpage, entonces se ignora la primera penalización, pero la segunda invoca al mecanismo de salida. Cuando una nota al pie de página es demasiado larga para caber completamente en una sola página, se puede dividir en dos o más partes. Cuando ocurre esta división y la primera parte de la nota al pie se muestra en una página, la segunda "página" se encarga de mostrar el resto de la nota al pie de página.

Functions used in the output routine

@outputpage

\@outputpage: Produce una página de salida o envía una página en donde se contiene al caja \␣

@outputbox como la parte de texto. También establece \@colht :=G \textheight. El estilo de la página se determina de la siguiente manera:

```
IF @thispagestyle = true
  THEN utilice el estilo \thispagestyle
  ELSE utilice el estilo de página ordinario.
```

Este macro es la que se encarga de maquetar el encabezado, el contenido o texto y el pie de página, y además asigna el estilo de página.

Definición de este macro a grandes rasgos:

1. \begingroup inicia un grupo local para limitar el alcance de ciertos cambios.
2. \let \protect \noexpand evita que ciertos comandos se expandan.
3. \language\document@default@language establece el idioma a utilizar en la página.
4. \@resetactivechars resetea la categoría de los caracteres activos.
5. \global\let\@@if@newlist\if@newlist \global\@newlistfalse hace una copia de la condición \if@newlist y luego la establece a false.
6. \@parboxrestore resetea los parámetros a sus valores por defecto.
7. \shipout \vbox{...} genera la página de salida. Dentro de este bloque, se establece el estilo de la página, se manejan los encabezados y pies de página, se colocan los márgenes, se establece el tamaño de fuente a su valor predeterminado, y finalmente, se coloca el contenido de \@outputbox en la página.
8. \global\let\if@newlist\@@if@newlist restaura el valor original de \if@newlist.
9. \global \@colht \textheight establece la altura de la columna al valor actual de \textheight.
10. \stepcounter{page} incrementa el contador de páginas.
11. \let\firstmark\botmark establece la primera marca en la página a la última marca en la página.

\@tryfcolumn\FLIST

\@tryfcolumn\FLIST trata de construir una columna de flotantes a partir de una lista de flotantes disponibles. Una explicación sería de la siguiente forma:

- \@tryfcolumn\FLIST: Este es el comando principal que intenta formar una columna de flotantes. \FLIST es una lista de flotantes disponibles que podrían utilizarse para llenar la columna.
- \@colht: Este es la altura total de la columna. La columna de flotantes se rellena hasta que su altura total alcance este valor.
- \@fpmin: Este es la altura mínima que un flotante debe tener para ser considerado en esta columna.

- \@fpsep: Este es el espacio que se coloca entre flotantes en la columna.
- \@fptop y \@fpbot: Estos son los espacios que se colocan en la parte superior e inferior de la columna, respectivamente.

Si el comando puede formar con éxito una columna de flotantes, realiza las siguientes acciones:

- Establece \@outputbox := L como la columna de flotantes que acaba de formar.
- Establece la variable @fcolmade en verdadero (\@fcolmade:=G true), indicando que ha logrado formar una columna de flotantes.
- Actualiza \FLIST para eliminar los flotantes que acaba de usar en la columna.

```
\FLIST := G \FLIST - flotantes colocados en caja
```

- Agrega los flotantes que acaba de usar a \@freelist, que es una lista de flotantes que ya se han colocado y ya no están disponibles.

```
\@freelist := G \@freelist + flotantes colocados en caja
```

Si el comando no puede formar una columna de flotantes, simplemente establece la variable @fcolmade en falso (\@fcolmade:=G false).

La macro de LaTeX \@makefcolumn es similar a \@tryfcolumn en que intenta formar una columna de objetos flotantes (como figuras o tablas) a partir de una lista de tales objetos. Sin embargo, su comportamiento difiere en lo que ocurre si no puede colocar todos los objetos flotantes en la columna.

\␣@makefcolumn

La macro toma un argumento, \FLIST, que es una lista de objetos flotantes. Si esta lista no está vacía, \@makefcolumn siempre creará una columna que contenga al menos el primer objeto flotante de la lista, incluso si este objeto flotante no cumple con el tamaño mínimo especificado por \@fpmin.

Funcionamiento del macro \@startcolumn :

\␣@startcolumn

1. Llama al macro \@tryfcolumn con \@deferlist como argumento. \@deferlist es una lista global de objetos flotantes que aún no han sido colocados en la página.
2. Si \@tryfcolumn devuelve con @fcolmade (una variable global que indica si se ha podido construir una columna de flotantes) establecido en false, entonces:
   - Establece globalmente \@toplist y \@botlist a los objetos flotantes de \@deferlist que deben ir en la parte superior e inferior de la columna, respectivamente, eliminándolos de \@deferlist. Esto lo hace utilizando \@colht como la altura total, los parámetros de estilo de página \@floatsep y \@textfloatsep, y los parámetros de colocación de flotantes \@topnum, \@toproom, \@botnum, \@botroom, \@colnum y \textfraction.
   - Establece globalmente \@colroom a \@colht menos la altura de los objetos flotantes añadidos. \@colroom indica la cantidad de espacio vertical que queda en la columna después de añadir los objetos flotantes.

@startdblcol-umn

Funcionamiento del macro \@startdblcolumn :

1. Llama a la macro \@tryfcolumn con \@dbldeferlist como argumento. \@dbldeferlist es una lista global de objetos flotantes que aún no han sido colocados en la página y que se deben extender a través de dos columnas.

2. Si \@tryfcolumn devuelve con @fcolmade (una variable global que indica si se ha podido construir una columna de flotantes) establecido en false, entonces:

   - Establece globalmente \@dbltoplist a los objetos flotantes de \@dbldeferlist que deben ir en la parte superior e inferior de la columna, respectivamente, eliminándolos de \@dbldeferlist. Esto lo hace utilizando \textheight como la altura total, y los parámetros de estilo de página como \@dblfloatsep, etc.
   - Establece globalmente \@colht a \textheight menos la altura de los objetos flotantes añadidos. \@colht indica la cantidad de espacio vertical que queda en la columna después de añadir los objetos flotantes.

\@combinefloats

La función interna de LaTeX \@combinefloats es responsable de combinar el texto principal de la página con los objetos flotantes que han sido asignados para aparecer en la parte superior (\@toplist) o en la parte inferior (\@botlist) de la página. El nuevo recuadro combinado se coloca en \@outputbox. Además, la función se encarga de manejar las listas de objetos flotantes y las listas de objetos flotantes libres.

Veamos en detalle lo que hace la función:

1. Combina el texto del cuadro \@outputbox con los objetos flotantes de \@toplist y \@botlist. \@outputbox es un cuadro que contiene el texto principal que se está construyendo para la página actual. \@toplist y \@botlist son listas de objetos flotantes que han sido asignados para aparecer en la parte superior e inferior de la página, respectivamente.

2. La nueva caja combinada se coloca en \@outputbox. Esto significa que después de que se llama a esta función, \@outputbox contendrá tanto el texto principal como los objetos flotantes especificados.

3. Utiliza \floatsep y \textfloatsep para las separaciones apropiadas. Estos son parámetros que determinan cuánto espacio debe haber entre diferentes objetos flotantes y entre objetos flotantes y texto.

4. Transfiere los elementos de \@toplist y \@botlist a \@freelist, y vacía estas listas. \@freelist es una lista de cajas de objetos flotantes que están disponibles para ser reutilizadas. Al mover los objetos flotantes de \@toplist y \@botlist a \@freelist, la función está indicando que estos objetos flotantes ya no necesitan ser considerados para la colocación en la página actual.

\@makecol

El macro \@makecol en LaTeX se encarga de crear una columna de contenido, incluyendo el texto principal, las notas al pie y los objetos flotantes (como figuras y tablas). El macro \@makecol toma el contenido de \box255 (la caja que generalmente contiene el contenido principal de la página), las notas al pie acumuladas,

y los flotantes en \@toplist y \@botlist (que son listas que contienen los flotantes destinados a aparecer en la parte superior e inferior de la página, respectivamente) y los combina en una única columna de altura \@colht. A menos que la altura de la página haya sido modificada localmente, \@colht es generalmente igual a la altura de la página.

El contenido combinado es colocado en la caja \@outputbox. Esto es lo que finalmente se "envía" para ser mostrado en la página cuando LaTeX decide que la página está lista para ser producida.

Las cajas en \@midlist (que es una lista que contiene flotantes destinados a aparecer en medio de la página) se colocan de nuevo en \@freelist. Esta acción recicla las cajas que no se necesitan, liberando memoria.

Finalmente, el valor de \maxdepth (que es la profundidad máxima que una caja puede tener) se restaura.

Explicación de la definición de \@makecol en el kernel de LaTeX:

1. \ifvoid\footins: Esta línea verifica si la caja de notas al pie (\footins) está vacía.
2. \setbox\@outputbox \box\@cclv: Si \footins está vacío, entonces el contenido de la caja \@cclv (que es otra notación para \box255, la caja que contiene el contenido principal de la página) se mueve a \@outputbox.
3. \setbox\@outputbox \vbox {...}: Si \footins no está vacío, entonces se crea una caja vertical que contiene tanto el contenido de \@cclv como las notas al pie. \unvbox \@cclv desempaqueta el contenido de \@cclv y \unvbox \footins desempaqueta las notas al pie.
4. \xdef\@freelist{\@freelist\@midlist}: Esto añade las cajas en \@midlist (flotantes que están programados para aparecer a mitad de la página) a \@freelist (una lista de cajas no utilizadas).
5. \global \let \@midlist \@empty: Esto vacía \@midlist.
6. \@combinefloats: Esto combina los flotantes que están programados para aparecer en la parte superior o inferior de la página con el contenido principal de la página.
7. \ifvbox\@kludgeins ... \@makespecialcolbox ...: Si la caja \@kludgeins es una caja vertical (es decir, si contiene algún material), entonces \@makespecialcolbox se usa para hacer un tipo especial de caja.
8. \setbox\@outputbox \vbox to\@colht {...}: Si \@kludgeins no es una caja vertical, entonces se crea una caja vertical de altura \@colht (la altura prevista de la columna).
9. \global \maxdepth \@maxdepth: Finalmente, \maxdepth (la profundidad máxima de la caja) se establece a \@maxdepth.

La macro \@opcol determina si se está utilizando un formato de una o dos columnas, y realiza diferentes acciones según el caso. \@opcol

En general, \@opcol genera una columna en donde este texto se coloca en la caja \@outputbox.

- Si \@twocolumn es false (indicando un formato de una sola columna), entonces llama a \@outputpage, establece \@colht igual a \textheight (es decir, configura la altura de la columna para que sea igual a la altura del texto), y luego llama a \@floatplacement (que configura la colocación de los flotantes en la página).
- Si \@twocolumn es true o verdadero (indicando un formato de dos columnas), entonces:
  - Si \@firstcolumn es true (indicando que esta es la primera de las dos columnas en la página), entonces coloca la caja \@outputbox en \@leftcolumn (almacenando el contenido de la columna en la caja para la columna izquierda) y establece \@firstcolumn en false.
  - Si \@firstcolumn es false (indicando que esta es la segunda de las dos columnas en la página), entonces genera la página actual de dos columnas, cualquier posible página de flotantes de dos columnas, y determina \@dbltoplist para la próxima página (configurando los flotantes que deben ir en la parte superior de la próxima página).

Código del macro:

```
\def \@opcol {%
  \if@twocolumn
    \@outputdblcol
  \else
    \@outputpage
  \fi
  \global \@mparbottom \z@ \global \@textfloatsheight \z@
  \@floatplacement
}
```

User commands that call or affect the output routine

```
\newpage == BEGIN \par\vfil\penalty -10000 END

\clearpage == BEGIN \newpage
                    \write -1{} % Part of hack to make sure no
                    \vbox{} % \write's get lost.
                    \penalty -10001
              END

\cleardoublepage == BEGIN \clearpage
                      if @twoside = true and c@page is even
                        then \hbox{} \newpage fi
                    END
```

En LaTeX, la macro \twocolumn[CUADRO] se utiliza para comenzar una nueva página en formato de dos columnas y colocar el contenido del CUADRO en la parte superior de la página. La sintaxis [CUADRO] es

opcional y se utiliza para proporcionar un contenido específico que se colocará en la parte superior de la página de dos columnas.

El `CUADRO` se coloca dentro de un parbox, que es una caja de texto con un ancho igual a `\textwidth`, lo que significa que abarcará toda la anchura disponible en la página de dos columnas.

Esta macro es especialmente útil cuando se desea mostrar títulos o encabezados que ocupen toda la anchura de las dos columnas en una página. Al utilizar `\twocolumn[CUADRO]`, se asegura que el contenido del CUADRO se coloque de manera adecuada en la parte superior de la página de dos columnas.

Además, hay una peculiaridad sorprendente asociada con esta macro. La extensión de `\@dbltextfloatsep`, que es la separación vertical entre los elementos flotantes de texto en la página de dos columnas, se insertará automáticamente entre el CUADRO y la parte superior de las dos columnas. Esto proporciona un espaciado adecuado entre el CUADRO y el contenido que le sigue en las dos columnas.

Float-handling mechanisms

Un número de inserción es simplemente un identificador único para cada flotante y @freelist es la lista de estos números que están actualmente disponibles para su uso.

El entorno de flotante obtiene un número de inserción B de la lista @freelist, coloca el flotante en la caja B y establece `\count B` a un ESPECIFICADOR DE FLOTANTE.

Para un flotante normal (de una sola columna), los salto de páginas de los flotantes se consigue de dos maneras:

- En modo horizontal externo: `\vadjust{\penalty -10002}`.
- En modo vertical: `\penalty -10003`.

Para un flotante de doble columna, coloca B en la lista `\@dbldeferlist`.

Para especificar un flotante existen dos especificaciones:

1. Una ESPECIFICACIÓN DE COLOCACIÓN (PLACEMENT SPECIFICATION) describe dónde puede colocarse el flotante.
2. Un TIPO (TYPE), que es una potencia de dos; por ejemplo, las figuras podrían ser flotantes de tipo 1, las tablas flotantes de tipo 2, los programas flotantes de tipo 4, etc.

El especificador de flotante se codifica de la siguiente manera, donde el bit 0 es el bit menos significativo.

LaTeX utiliza una serie de bits para codificar la información acerca de dónde puede colocarse un elemento flotante (como una figura o tabla) y qué tipo de flotante es. Cada bit en el especificador de flotante tiene un significado específico:

- El bit 0 indica si el flotante puede colocarse en la ubicación actual en el texto.

- El bit 1 indica si el flotante puede colocarse en la parte superior de una página.
- El bit 2 indica si el flotante puede colocarse en la parte inferior de una página.
- El bit 3 indica si el flotante puede colocarse en una página solo con flotantes (una página flotante).
- El bit 4 se establece en 1 a menos que la colocación incluya un !, que en LaTeX se utiliza para indicar que se deben ignorar ciertas restricciones en la colocación de los flotantes.
- El bit 5 se establece en 1 para indicar un flotante de tipo 1.
- El bit 6 se establece en 1 para indicar un flotante de tipo 2.

Por ejemplo, un flotante que puede colocarse en la parte superior de una página y es de tipo 1 tendría un especificador de flotante con el bit 1 y el bit 5 establecidos en 1.

Además, un especificador de flotante negativo se utiliza para indicar una nota al margen, que es un tipo de flotante que se coloca en el margen de la página.

Macros and data structures for processing floats

Una LISTA DE FLOTANTES es una serie de cajas `\boxa ... \boxN` y tiene la forma:

```
\@elt\boxa ... \@elt\boxN
```

donde `\boxI` se define por `\newinsert\boxI`.

Normalmente, `\@elt` es un `\let` a `\relax`, es decir, `\let\@elt\relax`.

Los siguientes macros se utilizan para manipular las listas de flotantes:

1. `\@next \CS \LIST {NONEMPTY}{EMPTY}`: Este macro toma una lista de flotantes y realiza una acción dependiendo de si la lista está vacía o no. Si la lista está vacía, se ejecuta el código bajo EMPTY. Si no está vacía, se asigna el primer flotante a `\CS`, se elimina ese flotante de la lista, y luego se ejecuta el código bajo NONEMPTY.
2. `\@bitor\NUM\LIST`: Este macro verifica si hay al menos un flotante en la lista `\LIST` que tiene el bit log2 `\NUM` en su especificador de flotante igual a 1. Si es así, la variable de cambio global @test se establece en verdadero. De lo contrario, permanece en falso. Por ejemplo, si quieres ver si hay algún flotante en `\LIST` que tenga el mismo tipo que el flotante I, ejecutarías `\@bitor` con `\NUM` igual a `[(\count I)/32]*32`.
3. `\@cons\LIST\NUM`: Este macro añade el número `\NUM` a la lista `\LIST`. Es decir, establece `\LIST` como `\LIST` seguido de `\@elt \NUM`.

Es importante recordar que estos son macros de LaTeX y son muy específicos para el manejo de flotantes y listas de flotantes en este contexto. Además, los nombres de las variables y funciones en los macros son convenciones utilizadas en LaTeX, y entender su significado y uso requiere un conocimiento profundo de este lenguaje de marcado.

CAPÍTULO 16

CLASES ESTÁNDARES

16.1 Módulos

Los siguientes módulos se utilizan en la implementación para dirigir a docstrip en la generación de archivos externos:

1. El módulo "article" produce la clase de documento "article".
2. El módulo "report" produce la clase de documento "report".
3. El módulo "size10" produce la opción de clase para tamaño de letra 10pt.
4. El módulo "size11" produce la opción de clase para tamaño de letra 11pt.
5. El módulo "size12" produce la opción de clase para tamaño de letra 12pt.
6. El módulo "book" produce la clase de documento "book".
7. El módulo "bk10" produce la opción de clase "book" para tamaño de letra 10pt.
8. El módulo "bk11" produce la opción de clase "book" para tamaño de letra 11pt.
9. El módulo "bk12" produce la opción de clase "book" para tamaño de letra 12pt.
10. El módulo "driver" produce un archivo de control de documentación.

En términos simples: Se están utilizando diferentes módulos para que "docstrip" genere archivos externos con diferentes características, como por ejemplo el tipo de clase de documento, tamaño de letra, etc.

16.2 Código inicial

En esta parte se definen algunos comandos que se utilizarán más adelante.

1. `\@ptsize`: Esta secuencia de control se utiliza para almacenar el segundo dígito del tamaño de letra que se está utilizando. Por lo tanto, normalmente su valor es uno de 0, 1 o 2.

```
<*article | report | book>
\newcommand\@ptsize{}
```

2. `\if@restonecol`: Cuando el documento se imprime en dos columnas, a veces es necesario cambiar temporalmente a una columna. Este cambio se utiliza para recordar volver a cambiar a dos columnas.

```
\newif\if@restonecol
```

3. `\if@titlepage`: Un interruptor para indicar si se debe producir una página de título. Para la clase de documento "article" el valor por defecto es no producir una página de título separada.

```
\newif\if@titlepage
<article>\@titlepagefalse
<!article>\@titlepagetrue
```

4. `\if@openright`: Un interruptor para indicar si los capítulos deben comenzar en una página a la derecha. El valor por defecto para la clase de documento "report" es no; para la clase "book" es sí.

```
<!article>\newif\if@openright
```

5. `\if@mainmatter`: El interruptor `\if@mainmatter`, solo disponible en la clase de documento "book", indica si se está procesando el material principal del libro.

```
<book>\newif\if@mainmatter \mainmattertrue
```

En términos simples: Se están definiendo varios interruptores que se utilizarán más adelante en el documento para controlar diferentes aspectos, como el tamaño de letra, si se debe imprimir en una o dos columnas, si se debe producir una página de título, etc.

16.3 Declaración de opciones

Tamaño de página

Las variables `\paperwidth` y `\paperheight` deben reflejar el tamaño físico del papel después de cortar. Para la impresión en una impresora de escritorio, este suele ser el tamaño real del papel, ya que no hay procesamiento posterior. Las clases para la producción real de libros probablemente agregarán otros tamaños de papel y además la producción de marcas de recorte.

En el modo de compatibilidad, algunas opciones están deshabilitadas porque no estaban presentes en versiones anteriores de LaTeX.

`\if@compatibility` trata de un modo de compatibilidad para LaTeX 2.09 que se activa si se utiliza la antigua instrucción `\documentstyle` en lugar de `\documentclass`. Está incluido para permitir compilar documentos de LaTeX 2.09 con LaTeX2e, lo cual era muy importante en el momento en que apareció, ya que la mayoría de los usuarios todavía tenían muchos documentos en LaTeX 2.09. Creo que esto ayudó a que la gente cambiara más rápidamente a la nueva versión. Si tienes que cambiar todos tus documentos existentes para poder compilarlos, pero luego tienes que cambiarlos de nuevo si tú o alguno de tus compañeros quiere utilizar el antiguo compilador, entonces piensas dos veces antes de actualizar.

```
\if@compatibility\else
  \DeclareOption{a4paper}
```

```
    {\setlength\paperheight {297mm}%
     \setlength\paperwidth {210mm}}
  \DeclareOption{a5paper}
    {\setlength\paperheight {210mm}%
     \setlength\paperwidth {148mm}}
  \DeclareOption{b5paper}
    {\setlength\paperheight {250mm}%
     \setlength\paperwidth {176mm}}
  \DeclareOption{letterpaper}
    {\setlength\paperheight {11in}%
     \setlength\paperwidth {8.5in}}
  \DeclareOption{legalpaper}
    {\setlength\paperheight {14in}%
     \setlength\paperwidth {8.5in}}
  \DeclareOption{executivepaper}
    {\setlength\paperheight {10.5in}%
     \setlength\paperwidth {7.25in}}
```

La opción landscape cambia los valores de `\paperheight` y `\paperwidth`, si se asume que el papel está en posición vertical.

```
  \DeclareOption{landscape}
    {\setlength\@tempdima{\paperwidth}%
     \setlength\paperwidth{\paperheight}%
     \setlength\paperheight{\@tempdima}}
\fi
```

Tamaño de letra

Las opciones de tamaño de tipografía se controlan definiendo la variable `\@ptsize` para contener el último dígito del tamaño en cuestión y haciendo una bifurcación en `\ifcase`. Esto se hace por motivos históricos, pero hace difícil declarar opciones de tamaño menores a 10pt. Sin embargo, es posible usar 9 y 8 si una clase no define ambas opciones de 8pt y 18pt.

```
\if@compatibility
  \renewcommand\@ptsize{10}
\else
  \DeclareOption{10pt}{\renewcommand\@ptsize{10}}
\fi
\DeclareOption{11pt}{\renewcommand\@ptsize{11}}
\DeclareOption{12pt}{\renewcommand\@ptsize{12}}
```

Impresión de una cara o de 2 caras

Para imprimir en ambos lados de la hoja, utilizamos el interruptor `\if@twoside`. También debemos establecer `\if@mparswitch` para que los párrafos de margen se coloquen en el margen exterior.

```
\if@compatibility\else
  \DeclareOption{oneside}{\@twosidefalse \@mparswitchfalse}
\fi
\DeclareOption{twoside}{\@twosidetrue \@mparswitchtrue}
```

Opciones de borrador

Si el usuario solicita el modo "draft" (borrador), se muestra cualquier caja que se desborde.

```
\DeclareOption{draft}{\setlength\overfullrule{5pt}}
\if@compatibility\else
  \DeclareOption{final}{\setlength\overfullrule{0pt}}
\fi
```

Opciones de portada o titlepage

Normalmente un artículo no tiene una página de título separada, pero si el usuario lo pide, puede tener una.

```
\DeclareOption{titlepage}{\@titlepagetrue}
\if@compatibility\else
  \DeclareOption{notitlepage}{\@titlepagefalse}
\fi
```

Openright option

Esta opción decide si un capítulo empieza en una página de la derecha o no.

```
(!article) \if@compatibility
(book)     \@openrighttrue
(!article) \else
(!article) \DeclareOption{openright}{\@openrighttrue}
(!article) \DeclareOption{openany}{\@openrightfalse}
(!article) \fi
```

Impresión de 2 columnas

La opción de imprimir en dos columnas o una sola columna se maneja a través de una herramienta de cambio.

```
\if@compatibility\else
  \DeclareOption{onecolumn}{\@twocolumnfalse}
\fi
\DeclareOption{twocolumn}{\@twocolumntrue}
```

Numeración de ecuaciones a la izquierda

La opción leqno se puede usar para tener los números de las ecuaciones en el lado izquierdo de la ecuación. Carga código que se genera automáticamente a partir de los archivos kernel cuando se crea el formato. Si el número de la ecuación recibe un formato especial, entonces en lugar de usar el archivo kernel, la clase tendrá que proporcionar el código de manera explícita.

```
\DeclareOption{leqno}{\input{leqno.clo}}
```

Flush left displays

La opción fleqn permite redefinir los entornos de matemáticas en pantalla de tal manera que se coloquen alineados a la izquierda, con una sangría de `\mathindent` desde el margen izquierdo actual. Carga código que se genera automáticamente a partir de los archivos del núcleo cuando se crea el formato.

```
\DeclareOption{fleqn}{\input{fleqn.clo}}
```

Open bibliography

La opción openbib produce el estilo de bibliografía "abierto", en el que cada bloque comienza en una nueva línea y las líneas siguientes en un bloque están sangradas por `\bibindent`.

```
\DeclareOption{openbib}{%
  \AtEndOfPackage{%
    \renewcommand\@openbib@code{%
      \advance\leftmargin\bibindent
      \itemindent -\bibindent
      \listparindent \itemindent
      \parsep \z@
    }%
    \renewcommand\newblock{\par}}%
}
```

16.4 Ejecución de opciones

Aquí se ejecutan las opciones predeterminadas para inicializar ciertas variables. Tenga en cuenta que la clase de documento ’libro’ siempre utiliza impresión a dos caras.

```
<*article>
\ExecuteOptions{letterpaper,10pt,oneside,onecolumn,final}
</article>
<*report>
\ExecuteOptions{letterpaper,10pt,oneside,onecolumn,final,openany}
</report>
<*book>
\ExecuteOptions{letterpaper,10pt,twoside,onecolumn,final,openright}
</book>
```

El comando `\ProcessOptions` ejecuta el código para cada opción FOO que se haya declarado y para la que el usuario haya escrito la opción FOO en su comando `\documentclass`. Para cada opción BAR que el usuario haya escrito, pero que no esté declarada, se asume que la opción es una opción global. Todas las opciones se pasarán como opciones del documento a cualquier comando `\usepackage` en el preámbulo del documento.

```
\ProcessOptions
```

Ahora que se han ejecutado todas las opciones, podemos cargar el archivo de opciones de clase elegido que contiene todo el código dependiente del tamaño. Este archivo contiene el código necesario para definir el tamaño y estilo de la página en base a las opciones seleccionadas por el usuario al crear su documento. Por ejemplo, si el usuario elige la opción "libro", el archivo de opciones de clase elegido incluirá el código para crear un formato de libro con dos caras de impresión.

```
<!book> \input{size1\@ptsize.clo}
<book> \input{bk1\@ptsize.clo}
</article | report | book>
```

16.5 Paquetes utilizado o cargados

Los archivos de clases estándar no cargan paquetes adicionales.

16.6 Diseño del documento

En esta sección analizaremos el manejo de tipografía.

Fuentes

LATEX proporciona al usuario comandos para cambiar el tamaño de la fuente, en relación con el tamaño principal. Cada comando de cambio de tamaño relativo, `\size`, ejecuta el comando `\@setfontsize\size<font-size><baselineskip>` donde:

1. `<font-size>` es el tamaño absoluto de la fuente a usar a partir de ahora.
2. `\baselineskip` es el valor normal de `\baselineskip` para el tamaño de la fuente seleccionada. (El valor real será `\baselinestretch*<baselineskip>`). Hay un número de comandos, definidos en el núcleo de LATEX, que acortan las siguientes definiciones y son utilizados en todo el documento. Estos son:

\@vpt	5	\@vipt	6	\@viipt	7
\@viiipt	8	\@ixpt	9	\@xpt	10
\@xipt	10.95	\@xiipt	12	\@xivpt	14.4

El comando de nivel de usuario para el tamaño principal es `\normalsize`. Internamente, LaTeX utiliza `\@normalsize` cuando se refiere al tamaño principal. `\@normalsize` se definirá para funcionar como `\normalsize` si este último se redefine desde su definición predeterminada (que solo emite un mensaje de error). De lo contrario, `\@normalsize` simplemente selecciona un tamaño de 10pt/12pt.

El macro `\normalsize` también establece nuevos valores para `\abovedisplayskip`, `\abovedisplayshortskip` y `\belowdisplayshortskip`.

```
<*10pt|11pt|12pt>
\renewcommand\normalsize{%
<*10pt>
\@setfontsize\normalsize\@xpt\@xiipt
\abovedisplayskip 10\p@ \@plus2\p@ \@minus5\p@
\abovedisplayshortskip \z@ \@plus3\p@
\belowdisplayshortskip 6\p@ \@plus3\p@ \@minus3\p@
</10pt>
<*11pt>
\@setfontsize\normalsize\@xipt{13.6}%
\abovedisplayskip 11\p@ \@plus3\p@ \@minus6\p@
\abovedisplayshortskip \z@ \@plus3\p@
\belowdisplayshortskip 6.5\p@ \@plus3.5\p@ \@minus3\p@
</11pt>
<*12pt>
\@setfontsize\normalsize\@xiipt{14.5}%
\abovedisplayskip 12\p@ \@plus3\p@ \@minus7\p@
\abovedisplayshortskip \z@ \@plus3\p@
\belowdisplayshortskip 6.5\p@ \@plus3.5\p@ \@minus3\p@
</12pt>
```

Esta afirmación se refiere a dos comandos en LATEX que controlan el espacio antes y después de una lista. \abovedisplayskip controla el espacio antes de la lista, mientras que \belowdisplayskip controla el espacio después de la lista. Según esta afirmación, estos dos comandos tienen valores iguales.

Además, se menciona que los parámetros para la lista de primer nivel son siempre proporcionados por \@listI. Este comando proporciona la configuración por defecto para una lista de primer nivel. Por lo tanto, esta afirmación significa que cualquier lista de primer nivel utilizará los valores por defecto proporcionados por \@listI.

```
    \belowdisplayskip \abovedisplayskip
    \let\@listi\@listI}
```

Llamamos a la fuente de tamaño normal.

```
  \normalsize
```

Utilizamos \MakeRobust en lugar de \DeclareRobustCommand anteriormente para evitar un registro en el registro por la redefinición.

```
  \ifx\MakeRobust\@undefined \else
    \MakeRobust\normalsize
  \fi
```

De manera similar a \normalsize declaramos \small.

```
  \DeclareRobustCommand\small{%
  <*10pt>
    \@setfontsize\small\@ixpt{11}%
    \abovedisplayskip 8.5\p@ \@plus3\p@ \@minus4\p@
    \abovedisplayshortskip \z@ \@plus2\p@
    \belowdisplayshortskip 4\p@ \@plus2\p@ \@minus2\p@
    \def\@listi{\leftmargin\leftmargini
                \topsep 4\p@ \@plus2\p@ \@minus2\p@
                \parsep 2\p@ \@plus\p@ \@minus\p@
                \itemsep \parsep}%
  </10pt>
  <*11pt>
    \@setfontsize\small\@xpt\@xiipt
    \abovedisplayskip 10\p@ \@plus2\p@ \@minus5\p@
    \abovedisplayshortskip \z@ \@plus3\p@
    \belowdisplayshortskip 6\p@ \@plus3\p@ \@minus3\p@
    \def\@listi{\leftmargin\leftmargini
                \topsep 6\p@ \@plus2\p@ \@minus2\p@
```

```
              \parsep 3\p@ \@plus2\p@ \@minus\p@
              \itemsep \parsep}%
</11pt>
<*12pt>
  \@setfontsize\small\@xipt{13.6}%
  \abovedisplayskip 11\p@ \@plus3\p@ \@minus6\p@
  \abovedisplayshortskip \z@ \@plus3\p@
  \belowdisplayshortskip 6.5\p@ \@plus3.5\p@ \@minus3\p@
  \def\@listi{\leftmargin\leftmargini
              \topsep 9\p@ \@plus3\p@ \@minus5\p@
              \parsep 4.5\p@ \@plus2\p@ \@minus\p@
              \itemsep \parsep}%
</12pt>
  \belowdisplayskip \abovedisplayskip
}
```

De manera similar a `\normalsize` declaramos `\footnotesize`.

```
\DeclareRobustCommand\footnotesize{%
<*10pt>
  \@setfontsize\footnotesize\@viiipt{9.5}%
  \abovedisplayskip 6\p@ \@plus2\p@ \@minus4\p@
  \abovedisplayshortskip \z@ \@plus\p@
  \belowdisplayshortskip 3\p@ \@plus\p@ \@minus2\p@
  \def\@listi{\leftmargin\leftmargini
              \topsep 3\p@ \@plus\p@ \@minus\p@
              \parsep 2\p@ \@plus\p@ \@minus\p@
              \itemsep \parsep}%
</10pt>
<*11pt>
  \@setfontsize\footnotesize\@viiipt{9.5}%
  \abovedisplayskip 8\p@ \@plus2\p@ \@minus4\p@
  \abovedisplayshortskip \z@ \@plus\p@
  \belowdisplayshortskip 4\p@ \@plus2\p@ \@minus2\p@
  \def\@listi{\leftmargin\leftmargini
              \topsep 4\p@ \@plus2\p@ \@minus2\p@
              \parsep 2\p@ \@plus\p@ \@minus\p@
              \itemsep \parsep}%
</11pt>
<*12pt>
  \@setfontsize\footnotesize\@viiipt{9.5}%
  \abovedisplayskip 9\p@ \@plus2\p@ \@minus5\p@
```

```
  \abovedisplayshortskip \z@ \@plus\p@
  \belowdisplayshortskip 4\p@ \@plus2\p@ \@minus2\p@
  \def\@listi{\leftmargin\leftmargini
              \topsep 4\p@ \@plus2\p@ \@minus2\p@
              \parsep 2\p@ \@plus\p@ \@minus\p@
              \itemsep \parsep}%
</12pt>
  \belowdisplayskip \abovedisplayskip
}
</10pt | 11pt | 12pt>
```

Estas macros son mucho más simples que las anteriores. Solo seleccionan un nuevo tamaño de fuente, pero no tocan los parámetros para las presentaciones y listas

```
<*10>
␣
↪  \DeclareRobustCommand\scriptsize{\@setfontsize\scriptsize\@viipt\@viiipt}
\DeclareRobustCommand\tiny{\@setfontsize\tiny\@vpt\@vipt}
\DeclareRobustCommand\large{\@setfontsize\large\@xiipt{14}}
\DeclareRobustCommand\Large{\@setfontsize\Large\@xivpt{18}}
\DeclareRobustCommand\LARGE{\@setfontsize\LARGE\@xviipt{22}}
\DeclareRobustCommand\huge{\@setfontsize\huge\@xxpt{25}}
\DeclareRobustCommand\Huge{\@setfontsize\Huge\@xxvpt{30}}
</10>
<*11>
\DeclareRobustCommand\scriptsize{\@setfontsize\scriptsize\@viiipt{9.5}}
\DeclareRobustCommand\tiny{\@setfontsize\tiny\@vipt\@viipt}
\DeclareRobustCommand\large{\@setfontsize\large\@xivpt{18}}
\DeclareRobustCommand\Large{\@setfontsize\Large\@xviipt{22}}
\DeclareRobustCommand\LARGE{\@setfontsize\LARGE\@xxpt{25}}
\DeclareRobustCommand\huge{\@setfontsize\huge\@xxvpt{30}}
\DeclareRobustCommand\Huge{\@setfontsize\Huge\@xxvpt{30}}
</11>
<*12>
\DeclareRobustCommand\scriptsize{\@setfontsize\scriptsize\@viiipt{9.5}}
\DeclareRobustCommand\tiny{\@setfontsize\tiny\@vipt\@viipt}
\DeclareRobustCommand\large{\@setfontsize\large\@xivpt{18}}
\DeclareRobustCommand\Large{\@setfontsize\Large\@xviipt{22}}
\DeclareRobustCommand\LARGE{\@setfontsize\LARGE\@xxpt{25}}
\DeclareRobustCommand\huge{\@setfontsize\huge\@xxvpt{30}}
\let\Huge=\huge
</12>
```

Párrafos

```
⟨*article | report | book⟩
\setlength\lineskip{1\p@}
\setlength\normallineskip{1\p@}
```

Esto se usa como un multiplicador para `\baselineskip`. El valor predeterminado es no estirar las líneas base. Tenga en cuenta que si este comando no se resuelve como "vacío", cualquier parte positiva o negativa en la especificación de `\baselineskip` se ignorará.

```
\renewcommand\baselinestretch{}
```

`\parskip` proporciona espacio vertical adicional entre párrafos `\parindent` es el ancho del sangrado del párrafo. El valor de `\parindent` depende de si estamos en modo de dos columnas.

```
\setlength\parskip{0\p@ \@plus \p@}
⟨/article | report | book⟩
⟨*10pt | 11pt | 12pt⟩
\if@twocolumn
  \setlength\parindent{1em}
\else
<10pt> \setlength\parindent{15\p@}
<11pt> \setlength\parindent{17\p@}
<12pt> \setlength\parindent{1.5em}
\fi
⟨/10pt | 11pt | 12pt⟩
```

Los valores de los parámetros `\smallskipamount`, `\medskipamount` y `\bigskipamount` se establecen en el núcleo de LATEX. Tal vez deban variar, dependiendo de la opción de tamaño especificada. Pero como siempre han tenido el mismo valor independientemente de la opción de tamaño, no los cambiamos para mantener la compatibilidad tanto con LATEX 2.09 como con versiones anteriores de LATEX2e.

```
⟨*10pt | 11pt | 12pt⟩
\setlength\smallskipamount{3\p@ \@plus 1\p@ \@minus 1\p@}
\setlength\medskipamount{6\p@ \@plus 2\p@ \@minus 2\p@}
\setlength\bigskipamount{12\p@ \@plus 4\p@ \@minus 4\p@}
⟨/10pt | 11pt | 12pt⟩
```

Los comandos `\nopagebreak` y `\nolinebreak` ponen penalizaciones para desalentar estos salto de línea en el punto en que se insertan. Usan `\@lowpenalty`, `\@medpenalty` o `\@highpenalty`, dependiendo de su argumento.

```
<*article|report|book>
\@lowpenalty 51
```

```
\@medpenalty 151
\@highpenalty 301
```

Penalizaciones para romper la página.

```
% \clubpenalty 150
% \widowpenalty 150
```

```
% \displaywidowpenalty 50
% \predisplaypenalty 10000
% \postdisplaypenalty 0
```

Page layout

Todas las dimensiones de los márgenes se miden desde un punto una pulgada de la parte superior e izquierda de la página, por lo que los cálculos respectivos se debe realizar con referencia a una pulgada de la parte superior e izquierda de la página.

Separación vertical

\headheight
\headsep
\topskip

El `\headheight` es la altura del cuadro que contendrá el encabezado corriente. El `\headsep` es la distancia entre la parte inferior del encabezado corriente y la parte superior del texto. El `\topskip` es la `\baselineskip` para la primera línea en una página. LATEX no funcionará correctamente si tiene un valor de `0pt`, Pues lo que no se recomienda realizar tal operación.

```
<*10pt | 11pt | 12pt>
\setlength\headheight{12\p@}
<!bk>\setlength\headsep{25\p@}
<10pt & bk>\setlength\headsep{.25in}
<11pt & bk>\setlength\headsep{.275in}
<12pt & bk>\setlength\headsep{.275in}
<10pt>\setlength\topskip{10\p@}
<11pt>\setlength\topskip{11\p@}
<12pt>\setlength\topskip{12\p@}
```

El `\footskip` es la distancia desde la línea base de la caja que contiene el pie de página hasta la línea base de la última línea de texto. En otras palabras, controla la distancia entre el pie de página y el último renglón de texto.

```
<!bk> \setlength\footskip{30\p@}
<10pt & bk> \setlength\footskip{.35in}
<11pt & bk> \setlength\footskip{.38in}
<12pt & bk> \setlength\footskip{30\p@}
```

\maxdepth y \@maxdepth son registros en TEX (similares a \topskip) que controlan la distancia entre la última línea de texto y la parte inferior de la página. El valor de \maxdepth es de 4pt en TeXplain y LaTeX 2.09. El valor de \maxdepth en LaTeX 2ε depende del tamaño de letra dado por \maxdepth + \topskip = typesize x 1.5. El objetivo es que \@maxdepth siempre tenga una copia de \maxdepth y que la distancia entre la última línea de texto y la parte inferior de la página sea adecuada.

```
\if@compatibility \setlength\maxdepth{4\p@} \else
\setlength\maxdepth{.5\topskip} \fi
```

Dimensión del texto

El \textwidth controla el ancho del área de texto que se imprime.

```
\if@compatibility
  \if@twocolumn
    \setlength\textwidth{410\p@}
  \else
<10pt&!bk> \setlength\textwidth{345\p@}
<11pt&!bk> \setlength\textwidth{360\p@}
<12pt&!bk> \setlength\textwidth{390\p@}
<10pt & bk> \setlength\textwidth{4.5in}
<11pt & bk> \setlength\textwidth{5in}
<12pt & bk> \setlength\textwidth{5in}
  \fi
```

Cuando no estamos en modo de compatibilidad con LaTeX 2.09, podemos establecer algunas de las dimensiones de manera diferente

```
\else
```

En primer lugar, calculamos el ancho máximo de texto, que permitiremos en el papel seleccionado y lo almacenamos en \@tempdima. Luego almacenamos la longitud de una línea con aproximadamente 60 a 70 caracteres en \@tempdimb. Los valores dados son más o menos adecuados cuando se usan las fuentes Computer Modern.

```
  \setlength\@tempdima{\paperwidth}
  \addtolength\@tempdima{-2in}
<10pt>  \setlength\@tempdimb{345\p@}
<11pt>  \setlength\@tempdimb{360\p@}
<12pt>  \setlength\@tempdimb{390\p@}
```

Ahora podemos establecer el ancho del texto, dependiendo de si usaremos una o dos columnas. En el modo de dos columnas, cada columna no debe ser más ancha que \@tempdimb (lo que podría suceder en un papel a3, por ejemplo).

```
    \if@twocolumn
      \ifdim\@tempdima>2\@tempdimb\relax
        \setlength\textwidth{2\@tempdimb}
      \else
        \setlength\textwidth{\@tempdima}
      \fi
```

En modo de una sola columna, el texto no debe ser más ancho que el mínimo de la anchura del papel (menos 2 pulgadas para los márgenes) y la longitud máxima de una línea definida por el número de caracteres.

```
    \else
      \ifdim\@tempdima>\@tempdimb\relax
        \setlength\textwidth{\@tempdimb}
      \else
        \setlength\textwidth{\@tempdima}
      \fi
    \fi
  \fi
```

Aquí modificamos el ancho del texto un poco para que sea un número entero de puntos.

```
  \if@compatibility\else
    \@settopoint\textwidth
  \fi
```

\textheight

```
  \if@compatibility
  ⟨10pt&!bk⟩ \setlength\textheight{43\baselineskip}
  ⟨10pt & bk⟩ \setlength\textheight{41\baselineskip}
  ⟨11pt⟩ \setlength\textheight{38\baselineskip}
  ⟨12pt⟩ \setlength\textheight{36\baselineskip}
```

```
  \else
    \setlength\@tempdima{\paperheight}
    \addtolength\@tempdima{-2in}
    \addtolength\@tempdima{-1.5in}
    \divide\@tempdima\baselineskip
    \@tempcnta=\@tempdima
    \setlength\textheight{\@tempcnta\baselineskip}
  \fi
```

```
  \addtolength\textheight{\topskip}
```

Márgenes

Aquí se modifican los valores de los parámetros de márgenes, basándose en el tamaño del papel que se está usando. En los cálculos, se tiene en cuenta el `\marginparsep`, por lo que primero se le da su valor.

El espacio horizontal entre el texto principal y las notas marginales se determina por `\marginparsep`, y la separación vertical mínima entre dos notas marginales se controla por `\marginparpush`.

```
\if@twocolumn
  \setlength\marginparsep{10\p@}
\else
<10pt&!bk> \setlength\marginparsep{11\p@}
<11pt&!bk> \setlength\marginparsep{10\p@}
<12pt&!bk> \setlength\marginparsep{10\p@}
<bk> \setlength\marginparsep{7\p@}
\fi
<10pt | 11pt> \setlength\marginparpush{5\p@}
<12pt> \setlength\marginparpush{7\p@}
```

Notas en pie de página

Todos los macros de las notas de pie de página ya se han explicado anteriormente, por lo tanto no se va a explicar.

`\footnotesep`

```
⟨10pt⟩\setlength\footnotesep{6.65\p@}
⟨11pt⟩\setlength\footnotesep{7.7\p@}
⟨12pt⟩\setlength\footnotesep{8.4\p@}
```

`\footins`

```
⟨10pt⟩\setlength{\skip\footins}{9\p@ \@plus 4\p@ \@minus 2\p@}
⟨11pt⟩\setlength{\skip\footins}{10\p@ \@plus 4\p@ \@minus 2\p@}
⟨12pt⟩\setlength{\skip\footins}{10.8\p@ \@plus 4\p@ \@minus 2\p@}
⟨/10pt | 11pt | 12pt⟩
```

Float placement parameters

En TeX y LaTeX, un parámetro es una variable que se utiliza para almacenar valores que controlan el comportamiento de ciertos comandos. Los parámetros pueden ser de diferentes tipos, como enteros (counters) o flotantes.

En el núcleo o kernel LaTeX 2_ε, todos los parámetros de tipo flotante tienen valores predeterminados. Esto significa que ya tienen un valor asignado por defecto. Sin embargo, si queremos cambiar ese valor, debemos hacerlo explícitamente utilizando el comando `\renewcommand`.

\c@topnumber: El contador topnumber almacena el número máximo de flotantes que pueden aparecer en la parte superior de una página de texto.

```
<*article|report|book>
\setcounter{topnumber}{2}
```

\topfraction es un parámetro LaTeX que controla la cantidad de espacio en la parte superior de una página que puede ser ocupado por flotantes. Este parámetro se utiliza para asegurarse de que los flotantes no ocupen una cantidad excesiva de espacio en la parte superior de la página, lo que podría interferir con la legibilidad del texto. El valor de \topfraction se especifica en términos de una fracción de la página total, por lo que si \topfraction se establece en 0.7, significa que los flotantes en la parte superior pueden ocupar hasta el 70% de la página.

```
\renewcommand\topfraction{0.7}
```

El contador \c@bottomnumber permite controlar la presentación visual del documento, especificando el número máximo de flotantes que pueden aparecer en la parte inferior de una página. De esta manera, se puede asegurar que los flotantes no interrumpan la lectura del texto y la presentación del documento sea clara y ordenada.

```
\setcounter{bottomnumber}{1}
```

\bottomfraction es un parámetro utilizado para controlar la cantidad de espacio en la parte inferior de una página que puede ser ocupado por flotantes. Este parámetro es esencial para garantizar que los flotantes no interrumpan la lectura del texto y que la presentación del documento sea clara y ordenada. El valor de \bottomfraction se especifica en términos de una fracción de la página total, por lo que si \bottomfraction se establece en 0.3, significa que los flotantes en la parte inferior pueden ocupar hasta el 30% de la página.

```
\renewcommand\bottomfraction{0.3}
```

\c@totalnumber es un contador utilizado para controlar el número total de flotantes que pueden aparecer en cualquier página de texto.

```
\setcounter{totalnumber}{3}
```

\textfraction: Este parámetro indica la parte mínima de una página de texto que debe estar ocupada por texto. El valor de 0.2se especifica en términos de una fracción de la página total, por lo que si \textfraction se establece en 0.2, significa que al menos el 20% de la página debe estar ocupado por texto.

```
\renewcommand\textfraction{.2}
```

`\floatpagefraction`: Este parámetro indica la parte mínima de una página que debe estar ocupada por objetos flotantes antes de producir una "página flotante".

```
\renewcommand\floatpagefraction{.5}
```

`\c@dbltopnumber` es un contador que se utiliza para controlar el número máximo de objetos flotantes en dos columnas que pueden aparecer en la parte superior de una página de texto en dos columnas.

```
\setcounter{dbltopnumber}{2}
```

`\dbltopfraction` es un parámetro que indica la fracción máxima de una página de texto en dos columnas que puede ser ocupada por objetos flotantes en dos columnas en la parte superior.

```
\renewcommand\dbltopfraction{.7}
```

`\dblfloatpagefraction` es un parámetro que indica la fracción mínima de una página que debe ser ocupada por objetos flotantes de dos columnas antes de que se produzca una "página flotante".

```
\renewcommand\dblfloatpagefraction{.5}
</article|report|book>
```

El comando `\floatsep` es el espacio vertical que se deja entre los flotantes adyacentes en una página. El comando `\textfloatsep` es el espacio entre el texto principal y los flotantes en la parte superior o inferior de la página. El comando `\intextsep` es el espacio entre los flotantes dentro del texto y el texto en sí.

`\floatsep`

`\textfloatsep`

`\intextsep`

```
⟨*10pt⟩
\setlength\floatsep{12\p@ \@plus 2\p@ \@minus 2\p@}
\setlength\textfloatsep{20\p@ \@plus 2\p@ \@minus 4\p@}
\setlength\intextsep{12\p@ \@plus 2\p@ \@minus 2\p@}
⟨/10pt⟩
⟨*11pt⟩
\setlength\floatsep{12\p@ \@plus 2\p@ \@minus 2\p@}
\setlength\textfloatsep{20\p@ \@plus 2\p@ \@minus 4\p@}
\setlength\intextsep{12\p@ \@plus 2\p@ \@minus 2\p@}
⟨/11pt⟩
⟨*12pt⟩
\setlength\floatsep{12\p@ \@plus 2\p@ \@minus 4\p@}
\setlength\textfloatsep{20\p@ \@plus 2\p@ \@minus 4\p@}
\setlength\intextsep{14\p@ \@plus 4\p@ \@minus 4\p@}
⟨/12pt⟩
```

Aquellos objetos flotantes que ocupan todo el ancho de área de texto, `\textwidth`, se coloca en una página de texto, mientras que en modo de 2 columnas la separación entre flotantes está controlado por `\dblfloatsep` y `\dbltextfloatsep`.

\dblfloatsep

\dbltextfloatsep

El comando \dblfloatsep es el espacio entre flotantes adyacentes que se mueven hacia la parte superior o inferior de la página de texto. El comando \dbltextfloatsep es el espacio entre el texto principal y los flotantes en la parte superior o inferior de la página.

```
<*10pt>
\setlength\dblfloatsep{12\p@ \@plus 2\p@ \@minus 2\p@}
\setlength\dbltextfloatsep{20\p@ \@plus 2\p@ \@minus 4\p@}
</10pt>
<*11pt>
\setlength\dblfloatsep{12\p@ \@plus 2\p@ \@minus 2\p@}
\setlength\dbltextfloatsep{20\p@ \@plus 2\p@ \@minus 4\p@}
</11pt>
<*12pt>
\setlength\dblfloatsep{14\p@ \@plus 2\p@ \@minus 4\p@}
\setlength\dbltextfloatsep{20\p@ \@plus 2\p@ \@minus 4\p@}
</12pt>
```

\@fptop

\@fpsep

\@fpbot

Cuando los objetos flotantes se separan en una página o páginas separadas, la página se controla por los siguientes macros \@fptop, \@fpsep y \@fpbot. El comando \@fptop controla el espacio entre la parte superior de la página y el objeto flotante. El comando \@fpsep es el espacio adyacente entre objetos flotantes. El comando \@fpbot es el espacio entre la página inferior y el objeto flotante. Estes comando se utilizan para modo de una columna y modo de 2 columnas.

Modo de una columna:

```
<*10pt>
\setlength\@fptop{0\p@ \@plus 1fil}
\setlength\@fpsep{8\p@ \@plus 2fil}
\setlength\@fpbot{0\p@ \@plus 1fil}
</10pt>
<*11pt>
\setlength\@fptop{0\p@ \@plus 1fil}
\setlength\@fpsep{8\p@ \@plus 2fil}
\setlength\@fpbot{0\p@ \@plus 1fil}
</11pt>
<*12pt>
\setlength\@fptop{0\p@ \@plus 1fil}
\setlength\@fpsep{10\p@ \@plus 2fil}
\setlength\@fpbot{0\p@ \@plus 1fil}
</12pt>
```

Modo de 2 columnas:

```
<*10pt>
\setlength\@dblfptop{0\p@ \@plus 1fil}
\setlength\@dblfpsep{8\p@ \@plus 2fil}
\setlength\@dblfpbot{0\p@ \@plus 1fil}
</10pt>
<*11pt>
\setlength\@dblfptop{0\p@ \@plus 1fil}
\setlength\@dblfpsep{8\p@ \@plus 2fil}
\setlength\@dblfpbot{0\p@ \@plus 1fil}
</11pt>
<*12pt>
\setlength\@dblfptop{0\p@ \@plus 1fil}
\setlength\@dblfpsep{10\p@ \@plus 2fil}
\setlength\@dblfpbot{0\p@ \@plus 1fil}
</12pt>
<*article | report | book>
```

Estilo de página

El estilo de página “foo” se define mediante la definición del comando `\ps@foo`. Este comando solo debería hacer definiciones locales. No debería haber espacios sueltos en la definición, ya que podrían causar espacios adicionales.

El comando `\ps@...` define los macros `\@oddhead`, `\@oddfoot`, `\@evenhead` y `\@evenfoot` para definir los encabezados y pies de página, por ejemplo, `\@oddhead` es la macro que produce el contenido de la caja de encabezado para las páginas con números impares. Se llama dentro de un `\hbox` con un ancho de `\textwidth`.

Convensión de marcado

El estilo de las páginas se definen con los comandos `\chaptermark`, `\sectionmark`, etc. que son llamados por otros comandos como `\chapter` para establecer una marca.

LATEX extiende la funcionalidad `\mark` de TEX produciendo dos tipos de marcas: una marca “izquierda” y una marca “derecha”. Estas marcas se pueden agregar mediante los comandos:

1. `\markboth{<LEFT>}{<RIGHT>}`: Agrega ambas marcas.

2. `\markright{<RIGHT>}`: Agrega una marca a la derecha.

3. `\leftmark`: Se utiliza en las macros `\@oddhead`, `\@oddfoot`, `\@evenhead` o `\@evenfoot` y obtiene la marca actual de la izquierda. Funciona de manera similar al comando `\botmark` de TEX.

 `\rightmark`: Se utiliza en las macros `\@oddhead`, `\@oddfoot`, `\@evenhead` o `\@evenfoot` y obtiene la marca actual de la derecha. Funciona de manera similar al comando `\firstmark` de TEX.

La marca izquierda cambia con un comando `\chapter` y la marca derecha cambia con un comando `\section`. Sin embargo, si dos `\markboth` ocurren en la misma página, producen resultados algo anómalos. Comandos como `\tableofcontents` que deben establecer las marcas en algunos estilos de página utilizan un comando `\@mkboth`, que es `\let` por el comando de estilo de página (`\ps@...`) a `\markboth` para establecer la cabecera o a `\@gobbletwo` para no hacer nada.

Definiendo estilo de página

En LaTeX los estilos de páginas definidos son `empty` y `plain`.

La definición del estilo de página "headings" (encabezados) tiene que ser diferente para la impresión a dos caras que para la impresión a una sola cara. Esto se debe a que en la impresión a dos caras se deben tener en cuenta factores adicionales, como la orientación de las páginas y la colocación de los encabezados, para garantizar una presentación adecuada en ambas caras de la hoja.

```
\if@twoside
  \def\ps@headings{%
```

El estilo de página en este caso tiene los pies de página vacíos y el encabezado contiene el número de página y una de las marcas. Esto significa que el encabezado de la página mostrará solo el número de página y una de las marcas establecidas, mientras que los pies de página estarán vacíos.

```
    \let\@oddfoot\@empty\let\@evenfoot\@empty
    \def\@evenhead{\thepage\hfil\slshape\leftmark}%
    \def\@oddhead{{\slshape\rightmark}\hfil\thepage}%
```

Cuando se utiliza este estilo de página, el contenido de la cabecera se determina por los títulos de capítulo y sección. Por lo tanto, hacemos que `\@mkboth` sea igual a `\markboth`.

```
    \let\@mkboth\markboth
```

En el documento de la clase “article”, se define el comando `\sectionmark` para borrar la marca de la derecha y colocar el número de la sección (cuando está numerada) y su título en la marca de la izquierda. La marca de la derecha se establece mediante el comando `\subsectionmark` para contener los títulos de las subsecciones.

Nota: el uso de `##1` para el parámetro del comando `\sectionmark` se definirá cuando se ejecute `\ps@headings`.

```
<*article>
    \def\sectionmark##1{%
```

```
        \markboth{\MakeUppercase{%
          \ifnum \c@secnumdepth >\z@
            \thesection\quad
          \fi
          ##1}}{}}%
      \def\subsectionmark##1{%
        \markright{%
          \ifnum \c@secnumdepth >\@ne
            \thesubsection\quad
          \fi
          ##1}}}
</article>
```

En las clases de documento report y book, utilizamos las macros `\chaptermark` y `\sectionmark` para llenar los encabezados.

Tenga en cuenta el uso de ##1 para el parámetro de la macro `\chaptermark`, que se definirá cuando se ejecute `\ps@headings`.

```
<*report|book>
    \def\chaptermark##1{%
      \markboth {\MakeUppercase{%
        \ifnum \c@secnumdepth >\m@ne
<book>    \if@mainmatter
            \@chapapp\ \thechapter. \ %
<book>    \fi
        \fi
        ##1}}{}}%
    \def\sectionmark##1{%
      \markright {\MakeUppercase{%
        \ifnum \c@secnumdepth >\z@
          \thesection. \ %
        \fi
        ##1}}}}
</report|book>
```

La definición de `\ps@headings` para impresión en un solo lado puede ser mucho más simple, ya que tratamos las páginas pares e impares de la misma manera. Por lo tanto, no necesitamos definir `\@even...`. Esto significa que la definición de estilo de página para impresión en un solo lado es más sencilla, ya que no hay diferencia entre las páginas pares e impares.

```
\else
  \def\ps@headings{%
```

```
    \let\@oddfoot\@empty
    \def\@oddhead{{\slshape\rightmark}\hfil\thepage}%
    \let\@mkboth\markboth
```

En la impresión en un solo lado, utilizamos `\markright` en lugar de `\markboth`, como lo hicimos en la impresión en dos lados. Esto significa que solo se aplica una marca "derecha" en lugar de marcas tanto a la izquierda como a la derecha en cada página.

```
<*article>
    \def\sectionmark##1{%
      \markright {\MakeUppercase{%
        \ifnum \c@secnumdepth >\m@ne
          \thesection\quad
        \fi
        ##1}}}}
</article>
<*report|book>
    \def\chaptermark##1{%
      \markright {\MakeUppercase{%
        \ifnum \c@secnumdepth >\m@ne
<book>    \if@mainmatter
            \@chapapp\ \thechapter. \ %
<book>    \fi
        \fi
        ##1}}}
    \def\sectionmark##1{%
      \markright {\MakeUppercase{%
        \ifnum \c@secnumdepth >\m@ne
          \thesection. \ %
        \fi
        ##1}}}}
</report|book>
\fi
```

En resumen, el estilo de página "myheadings" en LaTeX permite al usuario personalizar la cabecera corriente en su documento mediante el uso de los comandos `\markboth` y `\markright`, lo que le da un control total sobre lo que se incluirá en esta sección en la parte superior de cada página. Esto permite al usuario incluir información relevante y personalizar su documento de acuerdo a sus necesidades.

```
\def\ps@myheadings{%
  \let\@oddfoot\@empty\let\@evenfoot\@empty
  \def\@evenhead{\thepage\hfil\slshape\leftmark}%
  \def\@oddhead{{\slshape\rightmark}\hfil\thepage}%
```

Para que los comandos de marcado utilizados por los encabezados de capítulos y secciones estén desabilitados se debe utilizar los comandos de tipo \@gobble.

```
  \let\@mkboth\@gobbletwo
</article>  \let\chaptermark\@gobble
  \let\sectionmark\@gobble
</article>  \let\subsectionmark\@gobble
```

16.7 Marcado de un documento

El título

En esta subsección estudiaremos los diferentes parámetros que se ejecutan cuando invocamos el comando \maketitle. Estos parámetros pueden ser como el título del artículo o del libro y como de la misma forma el autor.

En el formato LATEXhay tres macros predefinidas (\title, \author y \date) que permiten al usuario introducir información sobre el título, autor o autores y fecha del documento. Esta información se almacena en variables internas que se llaman secuencias de control. Cuando el usuario utiliza el comando \maketitle, el formato utiliza la información almacenada en estas variables para crear una página de título para el documento.

Las definiciones de estas macros se proporcionan para que los usuarios puedan ver cómo se implementan internamente, pero no es necesario modificarlas para utilizar la función de título. El uso del comando \maketitle es suficiente para crear una página de título adecuada con la información proporcionada.

```
% \DeclareRobustCommand*{\title}[1]{\gdef\@title{#1}}
% \DeclareRobustCommand*{\author}[1]{\gdef\@author{#1}}
% \DeclareRobustCommand*{\date}[1]{\gdef\@date{#1}}
```

Como argumento obligatorio del macro o comando \date por defecto es:

```
% \date{\today}
```

Esto imprime la fecha de actual.

La definición del comando \maketitle depende de si se crea una página de título separada. Recordemos que en la clase de documentos report o book la separación de título en una hoja o página es automática o predeterminado y mientras tanto en un clase de documento article esto es opcional.

Para cambiar la apariencia de las notas en pie de página, debemos de definir localmente los comandos \footnotesize y \footnoterule para cambiar el aspecto de las notas al pie que se producen con el comando \thanks; estos cambios afectan a todas las notas al pie.

```
\if@titlepage
  \newcommand\maketitle{\begin{titlepage}%
```

```
\let\footnotesize\small
\let\footnoterule\relax
\let\footnote\thanks
```

En otras palabras, al hacer una página de título, se centra el título verticalmente en la página y esto se logra con el añadiendo el comando `\vskip`, lo que significa que está alineado en el centro de la página.

```
\null\vfil
\vskip 60\p@
```

El título se establece en una fuente grande (`\LARGE`) y se coloca en la parte superior de la página. Después, se deja un pequeño espacio y se establece el nombre del autor (o autores) en una fuente más pequeña (`\large`), todo dentro de un entorno de tabla para que queden en una única columna. Finalmente, antes de la fecha, se deja un poco de espacio en blanco para separar el nombre del autor y la fecha.

```
\begin{center}%
  {\LARGE \@title \par}%
  \vskip 3em%
  {\large
   \lineskip .75em%
    \begin{tabular}[t]{c}%
      \@author
    \end{tabular}\par}%
    \vskip 1.5em%
  {\large \@date \par}%       % Set date in \large size.
\end{center}\par
```

Cuando se crea una página de título, a menudo se incluye información adicional en una nota al pie de página. Para imprimir esta información, se llama al comando `\@thanks`, que se encarga de procesar y mostrar el contenido de la nota al pie. Después de imprimir la nota al pie, la página de título se considera completa y se cierra.

```
\@thanks
\vfil\null
\end{titlepage}%
```

Reiniciamos el contador de notas al pie de página, desactivamos `\thanks` y `\maketitle` y ahorraremos espacio de almacenamiento vaciando las macros de información interna.

```
\setcounter{footnote}{0}%
\global\let\thanks\relax
\global\let\maketitle\relax
\global\let\@thanks\@empty
```

```
      \global\let\@author\@empty
      \global\let\@date\@empty
      \global\let\@title\@empty
```

Después de establecer el título los comandos de declaración `\title`, etc. pueden desaparecer. La definición de `\and` solo tiene sentido dentro del argumento de `\author`, por lo que también puede eliminarse.

```
      \global\let\title\relax
      \global\let\author\relax
      \global\let\date\relax
      \global\let\and\relax
    }
  \else
```

Por lo tanto, primero comenzamos un nuevo grupo para mantener los cambios locales. Luego redefinimos `\thefootnote` para usar `\fnsymbol`; y cambiamos `\@makefnmark` para que las marcas de las notas al pie de página tengan una anchura cero (para mejorar la apariencia del centrado de los nombres de los autores).

```
  \else
    \newcommand\maketitle{\par
      \begingroup
        \renewcommand\thefootnote{\@fnsymbol\c@footnote}%
        \def\@makefnmark{\rlap{\@textsuperscript{\normalfont\@thefnmark}}}%
        \long\def\@makefntext##1{\parindent 1em\noindent
                \hb@xt@1.8em{%
                  \hss\@textsuperscript{\normalfont\@thefnmark}}##1}%
```

Si el documento está en formato de dos columnas, es posible que sea necesario iniciar una nueva página en modo de dos columnas y establecer el título en la anchura completa del texto para que tenga una apariencia coherente con el resto del contenido.

Sin embargo, la impresión real de la información del título se deja a `\@maketitle`, que es un comando utilizado para imprimir la información del título. De esta manera, se puede separar la presentación del título de la información real del título y manejar cada uno por separado.

En resumen, si el documento es de dos columnas, se debe iniciar una nueva página en modo de dos columnas y establecer el título en la anchura completa del texto, y que la impresión real de la información del título se deja al comando `\@maketitle`.

```
        \if@twocolumn
          \ifnum \col@number=\@ne
            \@maketitle
```

```
        \else
          \twocolumn[\@maketitle]%
        \fi
      \else
```

Si el documento no es de dos columnas, se debe iniciar una nueva página y evitar que los objetos flotantes aparezcan en la parte superior de la página, y luego se puede imprimir la información del título en la nueva página.

```
      \newpage
        \global\@topnum\z@   % Prevents figures from going at top of
         ↪  page.
        \@maketitle
      \fi
      \thispagestyle{plain}\@thanks
    \endgroup
    \setcounter{footnote}{0}%
    \global\let\thanks\relax
    \global\let\maketitle\relax
    \global\let\@maketitle\relax
    \global\let\@thanks\@empty
    \global\let\@author\@empty
    \global\let\@date\@empty
    \global\let\@title\@empty
    \global\let\title\relax
    \global\let\author\relax
    \global\let\date\relax
    \global\let\and\relax
  }
```

Cuando no se tiene una página de título separada, se inicia una nueva página y se deja un espacio en blanco para la información del título. Luego, se centra el título y se establece en una fuente `\LARGE` para resaltar su importancia.

Los nombres de los autores y la fecha se establecen en una fuente `\large` para dar un contraste adecuado con el tamaño del título y asegurar que toda la información sea legible y estéticamente agradable.

```
  \def\@maketitle{%
  \newpage
  \null
  \vskip 2em%
  \begin{center}%
  \let \footnote \thanks
```

```
      {\LARGE \@title \par}%
      \vskip 1.5em%
      {\large
        \lineskip .5em%
        \begin{tabular}[t]{c}%
          \@author
        \end{tabular}\par}%
      \vskip 1em%
      {\large \@date}%
    \end{center}%
    \par
    \vskip 1.5em}
  \fi
```

Capítulos y secciones

Building blocks

En LaTeX, se pueden definir nuevas secciones en el documento con la ayuda de la macro `\@startsection`. Esta macro toma seis argumentos requeridos y opcionalmente seguidos por un asterisco, un argumento opcional y un argumento requerido. Estos argumentos se utilizan para definir la apariencia de la sección, incluyendo su título, numeración, estilo de fuente, etc.

La definición de la macro `\@startsection` se utiliza en esta parte del archivo de clase para definir la apariencia de las diferentes secciones del documento, como capítulos, secciones, subsecciones, etc. Además, también se utiliza la macro `\secdef`, que se utiliza para definir secciones con una apariencia diferente para su uso en el encabezado o en la tabla de contenido.

```
\@startsection{⟨name⟩}{⟨level⟩}{⟨indent⟩}
{⟨beforeskip⟩}{⟨afterskip⟩}
{⟨style⟩}*[⟨altheading⟩]{⟨heading⟩}
```

En un archivo de clase hace uso de dos macros internas, `\@startsection` y `\secdef`. Para entender el funcionamiento de este comando, describimos su sintaxis:

1. `<name>`: El nombre del comando de nivel de usuario, por ejemplo, 'sección'.
2. `<level>`: Un número que denota la profundidad de la sección; por ejemplo, capítulo=1, sección=2, etc. Un número de sección se imprimirá si y sólo si `<level>` es menor que el valor del contador `secnumdepth`.
3. `<indent>`: La sangría del encabezado desde el margen izquierdo.
4. `<beforeskip>`: El valor absoluto de este argumento da el espacio que debe dejarse encima del encabezado. Si es negativo, se suprime la sangría de párrafo del texto que sigue al encabezado.

5. `<afterskip>`: Si es positivo, da el espacio que debe dejarse debajo del encabezado; de lo contrario, da el espacio que debe dejarse a la derecha de un encabezado que corre junto al texto.
6. `<style>`: Comandos para establecer el estilo del encabezado.
7. `*`: Cuando falta esto, el encabezado se numera y se incrementa el contador correspondiente.
8. `<altheading>`: Da un encabezado alternativo que se utilizará en la tabla de contenidos y en los encabezados de ejecución. Esto no debe estar presente cuando se utiliza la forma `*`.
9. `<heading>`: El encabezado de la nueva sección.

La macro `\secdef` se puede utilizar cuando se define un comando de sección sin usar `\@startsection` (recomiendo utilizar este macro cuando el estilo de la sección es compleja). Tiene dos argumentos:

```
\secdef⟨unstarcmds⟩⟨starcmds⟩
```

1. `<unstarcmds>` se utiliza para la forma normal del comando de sección.
2. `<starcmds>` se utiliza para el comando de sección con *.

El comando `\chapter` se define un clase de la siguiente manera:

```
\def\chapter { ... \secdef \CMDA \CMDB }
\def\CMDA[#1]#2{ ... }  % Command to define
                        % \chapter[...]{...}
\def\CMDB#1{ ... }  % Command to define
                    % \chapter*{...}
```

Comandos para marcado

La sección en la que se encuentran estas definiciones en el archivo de clase de LaTeX se refiere a las definiciones predeterminadas de estos comandos. La mayoría de ellos ya están definidos por el formato LaTeX, por lo que solo se muestran aquí como una referencia.

Los comandos `...mark` que se definen aquí incluyen, entre otros, `\chaptermark`, `\sectionmark`, `\subsectionmark` y `\subsubsectionmark`. Estos comandos se utilizan en la definición de los estilos de página para especificar qué información debe aparecer en los encabezados o pies de página de cada página del documento.

```
⟨!article⟩\newcommand*\chaptermark[1]{}
% \newcommand*\sectionmark[1]{}
% \newcommand*\subsectionmark[1]{}
% \newcommand*\subsubsectionmark[1]{}
% \newcommand*\paragraphmark[1]{}
% \newcommand*\subparagraphmark[1]{}
```

Definir contadores

El valor del contador `\secnumdepth` indica la profundidad de la sección de nivel más alto que producirá números de sección.

En algunos lenguajes de marcado como markdown, las secciones se utilizan para dividir un documento en partes más pequeñas y manejables al igual que en LaTeX. Las secciones pueden tener diferentes niveles de profundidad, como capítulos, secciones, subsecciones, etc. A menudo, se numeran las secciones para ayudar a los lectores a navegar por el documento y encontrar la información que necesitan.

En LaTeX, el contador `\secnumdepth` determina la profundidad de las secciones que deben numerarse. Por ejemplo, si el valor de `\secnumdepth` es 2, solo se numerarán las secciones de nivel 1 y 2, mientras que las de nivel 3 y superiores no tendrán números. Esto puede ser útil si desea evitar la numeración de secciones muy profundas o si desea tener un mayor control sobre la numeración de secciones en su documento.

```
⟨article⟩\setcounter{secnumdepth}{3}
⟨!article⟩\setcounter{secnumdepth}{2}
```

Estos contadores se utilizan para los números de sección. La macro

```
\newcounter{⟨newctr⟩}[⟨oldctr⟩]
```

define el contador `newctr`, que se reinicia a cero cuando se aumenta el contador `oldctr`. El contador `oldctr` ya debe estar definido.

Por ejemplo, si se desea contar el número de subsecciones dentro de cada sección en un documento, se puede crear un nuevo contador llamado “subsection” con la macro `\newcounter{subsection}[section]`. Esto significa que el contador “subsection” se reiniciará a cero cada vez que se aumente el contador de sección “section” . Por lo tanto, si hay tres subsecciones en la primera sección, sus números serán "1.1", "1.2" y "1.3", y si hay dos subsecciones en la segunda sección, sus números serán "2.1" y "2.2".

```
\newcounter {part}
<article> \newcounter {section}
<*report | book>
\newcounter {chapter}
\newcounter {section}[chapter]
</report | book>
\newcounter {subsection}[section]
\newcounter {subsubsection}[subsection]
\newcounter {paragraph}[subsubsection]
\newcounter {subparagraph}[paragraph]
```

Para cualquier contador CTR, `\theCTR` es una macro que define el formato impresa del contador CTR. Se define en términos de las siguientes macros:

1. `\arabic{COUNTER}`
2. `\roman{COUNTER}`
3. `\Roman{COUNTER}`
4. `\alph{COUNTER}`
5. `\Alph{COUNTER}`

```
\renewcommand\thepart{\@Roman\c@part}
<article>\renewcommand\thesection{\thesection\@arabic\c@section}
<*report | book>
\renewcommand\thechapter{\@arabic\c@chapter}
\renewcommand\thesection{\thechapter.\@arabic\c@section}
</report | book>
\renewcommand\thesubsection{\thesection.\@arabic\c@subsection}
\renewcommand\thesubsubsection{\thesubsection.\@arabic\c@subsubsection}
\renewcommand\theparagraph{\thesubsubsection.\@arabic\c@paragraph}
\renewcommand\thesubparagraph{\theparagraph.\@arabic\c@subparagraph}
```

El macro `\@chapapp` se define como `\chaptername` y el comando `\appendix` es redefinido a `\appendixname`.

```
⟨report | book⟩\newcommand\@chapapp{\chaptername}
```

Frontmatter, mainmatter y backmatter

En LaTeX, los libros y otros documentos grandes a menudo se dividen en varias secciones lógicas, como la introducción, el cuerpo del libro y la conclusión. El interruptor `\@mainmatter` se utiliza para determinar si se está procesando la sección principal del documento. Si el interruptor es verdadero, el documento está en la sección principal y los capítulos se numerarán. Si el interruptor es falso, el documento no está en la sección principal y los capítulos no se numerarán.

Por lo tanto, si se desea imprimir un prólogo o introducción en un libro sin números de capítulo, se puede establecer el interruptor `\@mainmatter` en falso antes de comenzar la sección de introducción. Esto se puede hacer utilizando el siguiente comando:

```
\makeatletter
\@mainmatterfalse
\makeatother
```

Luego, cuando se llegue a la sección principal del libro, el interruptor `\@mainmatter` se establecerá en verdadero nuevamente y los capítulos se numerarán de manera normal. Esto permite al autor tener un mayor control sobre el formato del libro y personalizar la numeración de los capítulos y secciones.

El comando \frontmatter inicia la numeración de páginas romanas y desactiva la numeración de capítulos.

```
<*book>
\newcommand\frontmatter{%
% \if@openright
  \cleardoublepage
% \else
%   \clearpage
% \fi
\@mainmatterfalse
\pagenumbering{roman}}
```

El comando \mainmatter limpia la página, inicia la numeración de página arábiga comenzando de 1 y activa la numeración de capítulos.

```
\newcommand\mainmatter{%
% if@openright
  \cleardoublepage
% \else
%   \clearpage
% \fi
\@mainmattertrue
\pagenumbering{arabic}}
```

El comando \backmatter limpia la página, desactiva la numeración de capítulos y deja la numeración de página sin cambios.

```
\newcommand\backmatter{%
  \if@openright
    \cleardoublepage
  \else
    \clearpage
  \fi
  \@mainmatterfalse}
</book>
```

Partes

El comando para comenzar una nueva parte de un documento en LaTeX es \part{<title>}. Comienza un nuevo párrafo, agrega un poco de espacio en blanco, suprime la sangría del primer párrafo y utiliza \secdef.

En la clase de artículo, la definición de \part es bastante simple; comenzamos un nuevo párrafo, agregamos un poco de espacio en blanco, suprimimos la sangría del primer párrafo y hacemos uso de \secdef. Como en otros comandos de sección (@startsection en el kernel LaTeX 2ε), debemos verificar el interruptor \@noskipsec y forzar el modo horizontal si está configurado.

```
<*article>
\newcommand\part{%
  \if@noskipsec \leavevmode \fi
  \par
  \addvspace{4ex}%
  \@afterindentfalse
  \secdef\@part\@spart}
</article>
```

Para las clases report y book hacemos las cosas un poco diferentes. Comenzamos una nueva página (derecha) y usamos el estilo de página simple.

```
<*report | book>
\newcommand\part{%
\if@openright
  \cleardoublepage
\else
  \clearpage
\fi
```

Cuando estamos haciendo un documento de dos columnas, esta será una página de una columna. Usamos \@tempswa para recordar cambiar de vuelta a dos columnas.

```
\if@twocolumn
  \onecolumn
  \@tempswatrue
\else
  \@tempswafalse
\fi
```

Necesitamos una caja vacía para evitar que el pegamento de fil desaparezca.

```
\null\vfil
```

Aquí invocamos el comando \secdef.

```
\secdef\@part\@spart}
</report | book>
```

El macro \@part es el formato del título de la sección. Este macro se define de diferente manera para un artículo y libro o artículo.

Capítulos

\chapter

Un capítulo siempre debe de comenzar en una nueva página, por lo que se debe utilizar el comando \clearpage y se establece a un estilo de página, en este caso plain.

```
<*report|book>
\newcommand\chapter{\if@openright\cleardoublepage\else\clearpage\fi
                    \thispagestyle{plain}%
```

El capítulo no se debe de mover, por lo tanto hacemos que no aparezca nada de objetos flotantes en la parte superior de la página.

```
                    \global\@topnum\z@
```

Luego eliminamos la sangría del párrafo con la opción \@afterindentfalse e invocamos al macro \secdef para definir los macros para establecer el título de los capítulos.

```
                    \@afterindentfalse
                    \secdef\@chapter\@schapter}
```

\@chapter

El macro \@chapter Se invoca cuando en un documento LaTeX invocamos el comando \chapter{PRUEBA} y esto se ejecuta cuando secnumdepth es mayor que -1. Si \@mainmatter es verdadero (el booleano \@mainmatter es verdadero por default) entonces genera enumeración de capítulo.

```
\def\@chapter[#1]#2{\ifnum \c@secnumdepth >\m@ne
<book>                       \if@mainmatter
                         \refstepcounter{chapter}%
                         \typeout{\@chapapp\space\thechapter.}%
                         \addcontentsline{toc}{chapter}%
                                   {\protect\numberline{\thechapter}#1}%
<*book>
                       \else
                         \addcontentsline{toc}{chapter}{#1}%
                       \fi
</book>
                    \else
                      \addcontentsline{toc}{chapter}{#1}%
                    \fi
```

Una vez que se haya escrito el título en la entrada de tabla de contenidos (toc), se llama al macro \chaptermark pasando como argumento obligatorio el título y luego se agrega espacio en blanco a la lista de figuras y tablas.

```
                        \chaptermark{#1}%
                        \addtocontents{lof}{\protect\addvspace{10\p@}}%
                        \addtocontents{lot}{\protect\addvspace{10\p@}}%
```

llamamos al \@makechapterhead para generar el formato real del capítulo. Cuando se esté en modo de dos columnas (twocolumn), el capítulo se debe de generar de manera especial para que ocupe todo el \textwidth. Cuando estemos en modo de una columna llamamos a \@afterheading para suprimir la sangría.

```
                        \if@twocolumn
                          \@topnewpage[\@makechapterhead{#2}]%
                        \else
                          \@makechapterhead{#2}%
                          \@afterheading
                        \fi}
```

Se deja espacio en blanco y luego se crea un grupo en donde la sangría es de 0pt y el texto es justificado hacia izquierda restableciendo la fuente.

```
\def\@makechapterhead#1{%
  \vspace*{50\p@}%
  {\parindent \z@ \raggedright \normalfont
```

Posteriormente se verifica si se debe imprimir el número de capítulos. Si es así, dejamos un espacio en blanco entre el número del capítulo y el título.

```
    \ifnum \c@secnumdepth >\m@ne
<book>      \if@mainmatter
        \huge\bfseries \@chapapp\space \thechapter
        \par\nobreak
        \vskip 20\p@
<book>      \fi
    \fi
```

Establecemos el título en una fuente grande y negrita. Luego se evita que no salte a la siguiente página después del título y finalmente se deja un espacio en blanco.

```
    \interlinepenalty\@M
    \Huge \bfseries #1\par\nobreak
```

```
      \vskip 40\p@
    }}
```

\@schapter

El comando \@schapter se invoca cuando se llama al comando \chapter*{PRUEBA} en un documento LaTeX. Este comando no genera enumeraciones de capítulos y es similar que el comando \@chapter.

```
\def\@schapter#1{\if@twocolumn
                    \@topnewpage[\@makeschapterhead{#1}]%
                  \else
                    \@makeschapterhead{#1}%
                    \@afterheading
                  \fi}
```

@makeschapterhead

El comando \@makeschapterhead da formato al encabezado del título.

```
\def\@makeschapterhead#1{%
  \vspace*{50\p@}%
  {\parindent \z@ \raggedright
    \normalfont
    \interlinepenalty\@M
    \Huge \bfseries  #1\par\nobreak
    \vskip 40\p@
  }}
```

Encabezados de nivel inferior

Para crear secciones de nivel inferior se debe recurrir al comando \@startsection.

\section

```
\newcommand\section{\@startsection {section}{1}{\z@}%
                                   {-3.5ex \@plus -1ex \@minus -.2ex}%
                                   {2.3ex \@plus.2ex}%
                                   {\normalfont\Large\bfseries}}
```

\subsection

```
\newcommand\subsection{\@startsection{subsection}{2}{\z@}%
                                     {-3.25ex\@plus -1ex \@minus -.2ex}%
                                     {1.5ex \@plus .2ex}%
                                     {\normalfont\large\bfseries}}
```

\subsubsection

```
\newcommand\subsubsection{\@startsection{subsubsection}{3}{\z@}%
                                     {-3.25ex\@plus -1ex \@minus -.2ex}%
```

```
                                   {1.5ex \@plus .2ex}%
                                   {\normalfont\normalsize\bfseries}}
```

\paragraph

```
\newcommand\paragraph{\@startsection{paragraph}{4}{\z@}%
                                    {3.25ex \@plus1ex \@minus.2ex}%
                                    {-1em}%
                                    {\normalfont\normalsize\bfseries}}
```

\subparagraph

```
\newcommand\subparagraph{\@startsection{subparagraph}{5}{\parindent}%
                                       {3.25ex \@plus1ex \@minus .2ex}%
                                       {-1em}%
                                      {\normalfont\normalsize\bfseries}}
```

Listas

Parámetros generales de una lista

Los siguientes comandos se utilizan para establecer los valores predeterminados para los parámetros del entorno de lista en LaTeX.

Los valores predeterminados del entorno de lista se establecen de la siguiente manera:

Primero, se asignan los valores de 0pt a `\rightmargin`, `\listparindent` y `\itemindent`. Luego, para una lista de nivel K, se llama al comando `\@listK`, donde 'K' representa 'i', 'ii', ..., 'vi'. (Por ejemplo, `@listiii` se llama para una lista de tercer nivel). Por convención, `\@listK` debe establecer `\leftmargin` en `\leftmarginK`.

Cuando nos encontremos en modo de 2 columnas, los márgenes izquierda adoptan valores distintos.

```
\if@twocolumn
  \setlength\leftmargini  {2em}
\else
  \setlength\leftmargini  {2.5em}
\fi
```

`\leftmargini` es el margen izquierdo de una lista de primer nivel. `\leftmarginii` es el margen izquierdo de una lista de segundo nivel, etc. Establecemos el margen exterior `\leftmargini` a `\leftmargin`.

```
\leftmargin  \leftmargini
```

Establecemos los márgenes interiores para diferentes niveles de lista.

```
\setlength\leftmarginii  {2.2em}
\setlength\leftmarginiii {1.87em}
```

```
\setlength\leftmarginiv  {1.7em}
\if@twocolumn
  \setlength\leftmarginv  {.5em}
  \setlength\leftmarginvi {.5em}
\else
  \setlength\leftmarginv  {1em}
  \setlength\leftmarginvi {1em}
\fi
```

`\labelsep` es la distancia entre la etiqueta (label) y el texto de un elemento (item); `\labelwidth` es el ancho de la etiqueta.

labelsep

labelwidth

```
\setlength  \labelsep  {.5em}
\setlength  \labelwidth{\leftmargini}
\addtolength\labelwidth{-\labelsep}
```

Cuando se invoque un entorno de lista y se deje una línea en blanco antes del invocar el entorno, entonces se adiciona un espacio vertical `\partopsep`.

partopsep

```
⟨/article | report | book⟩
⟨10pt⟩\setlength\partopsep{2\p@ \@plus 1\p@ \@minus 1\p@}
⟨11pt⟩\setlength\partopsep{3\p@ \@plus 1\p@ \@minus 1\p@}
⟨12pt⟩\setlength\partopsep{3\p@ \@plus 2\p@ \@minus 2\p@}
```

Las penalizacion `\@beginparpenalty` y `\@endparpenalty` se insertan antes y después del entorno lista o párrafo. Los valores establecidos indican que se rompa con mayor facilidad las páginas en este punto. La penalización `\@itempenalty` se inserta entre elementos `\item`.

```
⟨*article | report | book⟩
\@beginparpenalty -\@lowpenalty
\@endparpenalty   -\@lowpenalty
\@itempenalty     -\@lowpenalty
⟨/article | report | book⟩
```

`\@listi` define los valores de `\leftmargin`, `\parsep`, `\topsep`, `\itemsep`, etc. para las listas que aparecen en el nivel superior. La definición de este macro cambia de acuerdo con el tamaño del fuente. Para guardar las dimensiones de este nivel se crea una nueva macro `\@listI` para contener una copia de `\@listi`.

```
<*10pt|11pt|12pt>
\def\@listi{\leftmargin\leftmargini
<*10pt>
  \parsep 4\p@ \@plus2\p@ \@minus\p@
```

```
 \topsep 8\p@ \@plus2\p@ \@minus4\p@
 \itemsep4\p@ \@plus2\p@ \@minus\p@}
</10pt>
<*11pt>
 \parsep 4.5\p@ \@plus2\p@ \@minus\p@
 \topsep 9\p@ \@plus3\p@ \@minus5\p@
 \itemsep4.5\p@ \@plus2\p@ \@minus\p@}
</11pt>
<*12pt>
 \parsep 5\p@ \@plus2.5\p@ \@minus\p@
 \topsep 10\p@ \@plus4\p@ \@minus6\p@
 \itemsep5\p@ \@plus2.5\p@ \@minus\p@}
</12pt>
\let\@listI\@listi
```

De la misma forma, se define para los demás niveles.

```
\def\@listii {\leftmargin\leftmarginii
              \labelwidth\leftmarginii
              \advance\labelwidth-\labelsep
⟨*10pt⟩
   \topsep 4\p@ \@plus2\p@ \@minus\p@
   \parsep 2\p@ \@plus\p@ \@minus\p@
⟨/10pt⟩
⟨*11pt⟩
   \topsep 4.5\p@ \@plus2\p@ \@minus\p@
   \parsep 2\p@ \@plus\p@ \@minus\p@
⟨/11pt⟩
⟨*12pt⟩
   \topsep 5\p@ \@plus2.5\p@ \@minus\p@
   \parsep 2.5\p@ \@plus\p@ \@minus\p@
⟨/12pt⟩
   \itemsep \parsep}
\def\@listiii{\leftmargin\leftmarginiii
              \labelwidth\leftmarginiii
              \advance\labelwidth-\labelsep
⟨10pt⟩   \topsep 2\p@ \@plus\p@\@minus\p@
⟨11pt⟩   \topsep 2\p@ \@plus\p@\@minus\p@
⟨12pt⟩   \topsep 2.5\p@\@plus\p@\@minus\p@
              \parsep \z@
              \partopsep \p@ \@plus\z@ \@minus\p@
              \itemsep \topsep}
```

```
\def\@listiv {\leftmargin\leftmarginiv
              \labelwidth\leftmarginiv
              \advance\labelwidth-\labelsep}
\def\@listv  {\leftmargin\leftmarginv
              \labelwidth\leftmarginv
              \advance\labelwidth-\labelsep}
\def\@listvi {\leftmargin\leftmarginvi
              \labelwidth\leftmarginvi
              \advance\labelwidth-\labelsep}
</10pt|11pt|12pt>
```

Enumerate

El entorno enumerate tiene los siguientes contadores: `enumi`, `enumii`, `enumiii` y `enumiv`. Estos contandores pueden modificarse sus formatos como ya se estudió.

Los entornos enumerate y itemize ya están definidos en formato de LaTeX

Representaciones:

```
<*article | report | book>
\renewcommand\theenumi{\@arabic\c@enumi}
\renewcommand\theenumii{\@alph\c@enumii}
\renewcommand\theenumiii{\@roman\c@enumiii}
\renewcommand\theenumiv{\@Alph\c@enumiv}
```

Etiquetas:

```
\newcommand\labelenumi{\theenumi.}
\newcommand\labelenumii{(\theenumii)}
\newcommand\labelenumiii{\theenumiii.}
\newcommand\labelenumiv{\theenumiv.}
```

Definiciones de la salida del comando `\ref`:

```
\renewcommand\p@enumii{\theenumi}
\renewcommand\p@enumiii{\theenumi(\theenumii)}
\renewcommand\p@enumiv{\p@enumiii\theenumiii}
```

Itemize

Las enumeraciones del entorno itemize se controla de la siguiente forma:

```
\newcommand\labelitemi  {\labelitemfont \textbullet}
\newcommand\labelitemii {\labelitemfont \bfseries \textendash}
```

```
\newcommand\labelitemiii{\labelitemfont \textasteriskcentered}
\newcommand\labelitemiv{ \labelitemfont \textperiodcentered}
```

El comando `\labelitemfont` hace que siempre utilice una fuente determinada sin importar su circundancia.

```
\renewcommand\labelitemfont
  {\normalfont\fontfamily{lmss}\selectfont}
\renewcommand\labelitemfont
  {\rmfamily\normalshape}
\newcommand\labelitemfont{\normalfont}
```

Description

El entorno `description` se define de la siguiente forma:

```
\newenvironment{description}
               {\list{}{\labelwidth\z@ \itemindent-\leftmargin
                        \let\makelabel\descriptionlabel}}
               {\endlist}
```

Para cambiar el formato de la etiqueta se debe redefinir `\descriptionlabel`:

```
\newcommand*\descriptionlabel[1]{\hspace\labelsep
                                \normalfont\bfseries #1}
```

Definición de nuevos entornos

Titlepage

```
\if@compatibility
  \newenvironment{titlepage}
    {%
      \cleardoublepage
      \if@twocolumn
        \@restonecoltrue\onecolumn
      \else
        \@restonecolfalse\newpage
      \fi
      \thispagestyle{empty}%
      \setcounter{page}\z@
    }%
    {\if@restonecol\twocolumn \else \newpage \fi
```

```
    }
\else
  \newenvironment{titlepage}
    {%
      \cleardoublepage
      \if@twocolumn
        \@restonecoltrue\onecolumn
      \else
        \@restonecolfalse\newpage
      \fi
      \thispagestyle{empty}%
      \setcounter{page}\@ne
    }%
    {\if@restonecol\twocolumn \else \newpage \fi
     \if@twoside\else
        \setcounter{page}\@ne
     \fi
    }
\fi
```

Este entorno es utilizado en los comandos como `\maketitle` que hace que comience una nueva página impar, finalizando el modo de dos columnas si está activo y hace cambios en el contador de la página. Este entorno podemos utilizar para crear una portada del ilibro.

Appendix

```
\newcommand\appendix{\par
  \setcounter{chapter}{0}%
  \setcounter{section}{0}%
  \gdef\@chapapp{\appendixname}%
  \gdef\thechapter{\@Alph\c@chapter}}
```

Configuración de parámetros para entornos existentes

Matrix and tabular

```
\setlength\arraycolsep{5\p@}
\setlength\tabcolsep{6\p@}
\setlength\arrayrulewidth{.4\p@}
\setlength\doublerulesep{2\p@}
```

Estos comandos estándares ya se estudiaron anteriormente.

Tabbing

```
\setlength\tabbingsep{\labelsep}
```

Este comando define el espacio al utilizar el comando \`.

Minipage

Cuando ingresa al entorno minipage se llama al comando \@minipagerestore. Este macro realiza las configuraciones respectivas para que tenga un comportamiento definido dentro del entorno minipage.

Equation and eqnarray

\theequation

El contador de ecuaciones se reinicia cada vez que se inicia un nuevo capítulo. La enumeración de ecuaciones está prefijado por el número del capítulo.

```
⟨article⟩\renewcommand \theequation {\@arabic\c@equation}
⟨*report|book⟩
\@addtoreset {equation}{chapter}
\renewcommand\theequation
  {\ifnum \c@chapter>\z@ \thechapter.\fi \@arabic\c@equation}
⟨/report|book⟩
```

\jot

```
%\setlength\jot{3pt}
```

Espacio entre líneas de ecuaciones.

\@eqnnum

El macro \@eqnnum define la apariencia o el formato de las numeraciones de las ecuaciones.

```
% \def\@eqnum{(\theequation)}
```

Objetos flotantes

Ahora defineremos los comandos que se utilizan para crear objetos flotantes como tablas, figuras, etc. Estos comandos son utilizados por los entornos table y figure.

Figure

Definiendo entorno flotante figure.

Contador de figuras de acuerdo con la clase article, report y book.

```
<*article>
\newcounter{figure}
\renewcommand \thefigure {\@arabic\c@figure}
</article>
<*report|book>
\newcounter{figure}[chapter]
\renewcommand \thefigure
  {\ifnum \c@chapter>\z@ \thechapter.\fi \@arabic\c@figure}
</report|book>
```

Parámetros de objetos flotantes.

```
\def\fps@figure{tbp}
\def\ftype@figure{1}
\def\ext@figure{lof}
\def\fnum@figure{\figurename\nobreakspace\thefigure}
```

Ahora definimos los entornos `figure` y `figure*`.

```
\newenvironment{figure}
               {\@float{figure}}
               {\end@float}
\newenvironment{figure*}
               {\@dblfloat{figure}}
               {\end@dblfloat}
```

Table

Las tablas se definen de igual manera que una figura.

Contador de tablas de acuerdo con la clase article, report y book.

```
<*article>
\newcounter{table}
\renewcommand\thetable{\@arabic\c@table}
</article>
<*report|book>
\newcounter{table}[chapter]
\renewcommand \thetable
  {\ifnum \c@chapter>\z@ \thechapter.\fi \@arabic\c@table}
</report|book>
```

Parámetros de objetos flotantes para tablas.

```
\def\fps@table{tbp}
\def\ftype@table{2}
\def\ext@table{lot}
\def\fnum@table{\tablename\nobreakspace\thetable}
```

Ahora definimos los entornos `table` y `table*`.

```
\newenvironment{table}
              {\@float{table}}
              {\end@float}
\newenvironment{table*}
               {\@dblfloat{table}}
               {\end@dblfloat}
```

Subtítulos o caption

Especificamos el espacio antes y después del invocar el comando `\caption`.

```
\newlength\abovecaptionskip
\newlength\belowcaptionskip
\setlength\abovecaptionskip{10\p@}
\setlength\belowcaptionskip{0\p@}
```

Definiendo `\@makecaption` en forma de párrafo.

```
\long\def\@makecaption#1#2{%
\vskip\abovecaptionskip
```

Creamos una caja para evaluar si el texto se ajusta al línea de texto.

```
\sbox\@tempboxa{#1: #2}%
```

Evaluamos la caja y de acuerdo con el resultado, tomamos una decisión. Si la condición es verdadera, entonces imprimimos `#1: #2\par`.

```
\ifdim \wd\@tempboxa >\hsize
  #1: #2\par
```

si la condición es falsa realizamos las siguientes operaciones:

```
\else
  \global \@minipagefalse
  \hb@xt@\hsize{\hfil\box\@tempboxa\hfil}%
\fi
\vskip\belowcaptionskip}
```

Cambio de fuente

Los comandos que se definirán ahora se pueden utilizar tanto en modo texto y en modo matemático.

```
\DeclareOldFontCommand{\rm}{\normalfont\rmfamily}{\mathrm}
\DeclareOldFontCommand{\sf}{\normalfont\sffamily}{\mathsf}
\DeclareOldFontCommand{\tt}{\normalfont\ttfamily}{\mathtt}
\DeclareOldFontCommand{\bf}{\normalfont\bfseries}{\mathbf}
\DeclareOldFontCommand{\it}{\normalfont\itshape}{\mathit}
\DeclareOldFontCommand{\sl}{\normalfont\slshape}{\@nomath\sl}
\DeclareOldFontCommand{\sc}{\normalfont\scshape}{\@nomath\sc}
```

Los comandos `\cal` y `\mit` se utilizan para cambiar la fuente en el modo matemático, pero si se intentan usar fuera de ese contexto, no tendrán ningún efecto. Debido a que el Nuevo Esquema de Selección de Fuentes ha cambiado su definición y ahora muestra advertencias al usar estos comandos, debemos redefinirlos de manera personalizada para que se comporten según nuestras necesidades.

```
\DeclareRobustCommand*\cal{\@fontswitch\relax\mathcal}
\DeclareRobustCommand*\mit{\@fontswitch\relax\mathnormal}
```

16.8 Referencias cruzadas

Tabla de contenido y otros

El comando `\section` escribe un comando `\contentsline{section}{<title>}{<page>}` en el archivo .toc, donde <title> contiene el contenido de la entrada y <page> es el número de página. Si se están numerando las secciones, entonces <title> será del tipo `\numberline{<num>}{<heading>}` donde <num> es el número producido por `\thesection`. Otros comandos de seccionamiento funcionan de manera similar.

Un comando de `\caption` en un entorno "figure" escribe `\contentsline{figure}{\numberline{<num>}{<caption>}}{<page>}` en el archivo .lof, donde <num> es el número producido por `\thefigure` y <caption> es la leyenda de la figura. Funciona de manera similar en un entorno "table".

El comando `\contentsline{<name>}` se expande a `\l@<name>`. Por lo tanto, para especificar la tabla de contenidos, debemos definir `\l@chapter`, `\l@section`, `\l@subsection`, ...; para especificar la lista de figuras, debemos definir `\l@figure`; y así sucesivamente. La mayoría de estos se pueden definir con el comando `\@dottedtocline`, que funciona de la siguiente manera:

```
\@dottedtocline{<level>}{<indent>}{<numwidth>}{<title>}{<page>}
```

1. `<level>` Se producirá una entrada solo si <level> menor o igual al valor del contador tocdepth. Nota, `\chapter` es de nivel 0, `\section` es de nivel 1, etc.
2. `<indent>` La sangría desde el margen izquierdo exterior del inicio de la línea de contenidos.

3. <numwidth> El ancho de un cuadro en el que irá el número de sección, si <title> incluye un comando \numberline.
4. <title> Contenido de entrada.
5. <page> Número de página.

El comando \@pnumwidth utiliza los siguientes tres parámetros, que se establecen con un \newcommand (de manera que se pueden usar em para hacerlos dependientes de la fuente).

1. \pnumwidth La anchura de una caja en la que se coloca el número de página.
2. \@tocrmarg El margen derecho para entradas de varias líneas. Se desea que \@tocrmarg => \@pnumwidth
3. \@dotsep Separación entre puntos, en unidades mu. Debería definirse como un número como 2 o 1.7.

Table of contents

\tableofcontents

Esta macro se utiliza para solicitar que LATEX produzca un índice o una tabla de contenidos. En las clases de documentos "report" y "book", los índices, las tablas de contenidos, las figuras, etc. siempre se establecen en modo de columna única.

```
\newcommand{\tableofcontents}{%
<*report|book>
  \if@twocolumn
    \@restonecoltrue\onecolumn
  \else
    \@restonecolfalse
  \fi
```

El título se establece utilizando el comando \chapter*, asegurándose de que el encabezado corriente, si es necesario, contenga la información correcta.

```
  \chapter*{\contentsname
</report|book>
<article>     \section*{\contentsname
```

El código para \@mkboth se coloca dentro del encabezado para evitar cualquier influencia en el espacio vertical después del encabezado (en algunos casos).

```
    \@mkboth{%
      \MakeUppercase\contentsname}{\MakeUppercase\contentsname}}%
```

La tabla de contenidos real se crea llamando a \@starttoc{toc}. Después de eso, restauramos el modo de dos columnas si es necesario.

```
    \@starttoc{toc}%
  <!article> \if@restonecol\twocolumn\fi
  }
```

\l @part

Cada comando de seccionamiento requiere un macro adicional para formatear su entrada en la tabla de contenidos, como se describe anteriormente. El macro para la entrada de las partes se define de una manera especial.

Primero, nos aseguramos de que si debe ocurrir un salto de página, ocurra antes de esta entrada. También se agrega un poco de espacio en blanco y se comienza un grupo para mantener los cambios locales.

```
  \newcommand*\l@part[2]{%
    \ifnum \c@tocdepth >-2\relax
  <article>     \addpenalty\@secpenalty
  <!article>     \addpenalty{-\@highpenalty}%
      \addvspace{2.25em \@plus\p@}%
```

El macro \numberline requiere que se almacene en el registro temporal de LATEX @tempdima el ancho de la caja que contiene el número de la parte. Por lo tanto, lo inicializamos allí aunque no lo usemos internamente, ya que el valor utilizado es bastante grande para que algo como \numberline{VIII} todavía funcione.

```
  \setlength\@tempdima{3em}%
  \begingroup
```

Establecemos \parindent en 0pt y usamos \rightskip para dejar suficiente espacio para los números de página. Para evitar mensajes de cajas llenas, \parfillskip se establece en un valor negativo.

```
    \parindent \z@ \rightskip \@pnumwidth
    \parfillskip -\@pnumwidth
```

Ahora podemos establecer la entrada, en una fuente grande y negrita. Nos aseguramos de salir del modo vertical, establecer el título de la parte y agregar el número de página, establecerlo ajustado a la derecha.

```
      {\leavevmode
        \large \bfseries #1\hfil
        \hb@xt@\@pnumwidth{\hss #2%
                          \kern-\p@\kern\p@}}\par
```

Prevenimos un salto de página inmediatamente después de esta entrada, pero usamos \everypar para restablecer el interruptor \if@nobreak. Finalmente, cerramos el grupo.

```
    \nobreak
      \if@compatibility
        \global\@nobreaktrue
        \everypar{\global\@nobreakfalse\everypar{}}%
      \fi
    \endgroup
  \fi}
```

\l @chapter

Este macro formatea las entradas en el índice de contenidos para los capítulos. Es muy similar a \l@part.

Primero nos aseguramos de que, si debería ocurrir un salto de página, ocurra antes de esta entrada. Además, se agrega un poco de espacio en blanco y se comienza un grupo para mantener los cambios locales.

```
<*report|book>
\newcommand*\l@chapter[2]{%
  \ifnum \c@tocdepth >\m@ne
    \addpenalty{-\@highpenalty}%
    \vskip 1.0em \@plus\p@
```

El macro \numberline requiere que el ancho del cuadro que contiene el número de parte se almacene en el registro auxiliar de LATEX \@tempdima. Por lo tanto, lo inicializamos allí aunque no lo usemos internamente \numberline (la posición y los valores parecen cuestionables pero no se pueden cambiar sin producir problemas de compatibilidad). Comenzamos un grupo y cambiamos algunos de los parámetros del párrafo (ver también el comentario en \l@part con respecto a \rightskip).

```
    \setlength\@tempdima{1.5em}%
    \begingroup
      \parindent \z@ \rightskip \@pnumwidth
      \parfillskip -\@pnumwidth
```

Luego dejamos el modo vertical y cambiamos a una fuente en negrita.

```
      \leavevmode \bfseries
```

Debido a que no usamos \numberline aquí, debemos hacer algunos ajustes finos "a mano" antes de poder establecer la entrada. Desalentamos pero no prohibimos un salto de página inmediatamente después de una entrada de capítulo.

```
      \advance\leftskip\@tempdima
      \hskip -\leftskip
      #1\nobreak\hfil
      \nobreak\hb@xt@\@pnumwidth{\hss #2%
        \kern-\p@\kern\p@}\par
```

```
      \penalty\@highpenalty
    \endgroup
  \fi}
</*report|book>}
```

\l @section

En la clase de documento artículo, la entrada en el índice de contenidos para las secciones se parece mucho a las entradas de capítulos en las clases de documento informe y libro.

Primero nos aseguramos de que, si debería ocurrir un salto de página, ocurra antes de esta entrada. Además, se agrega un poco de espacio en blanco y se comienza un grupo para mantener los cambios locales.

```
<*article>
\newcommand*\l@section[2]{%
  \ifnum \c@tocdepth >\z@
    \addpenalty\@secpenalty
    \addvspace{1.0em \@plus\p@}%
```

El macro `\numberline` requiere que el ancho del cuadro que contiene el número de parte se almacene en el registro auxiliar de LATEX @tempdima. Por lo tanto, lo ponemos allí. Comenzamos un grupo y cambiamos algunos de los parámetros del párrafo (ver también el comentario en `\l@part` sobre `\rightskip`).

```
    \setlength\@tempdima{1.5em}%
    \begingroup
      \parindent \z@ \rightskip \@pnumwidth
      \parfillskip -\@pnumwidth
```

Luego, abandonamos el modo vertical y cambiamos a una fuente en negrita.

```
      \leavevmode \bfseries
```

Debido a que no usamos `\numberline` aquí, debemos hacer algunos ajustes finos "a mano" antes de poder establecer la entrada. Desalentamos pero no prohibimos un salto de página inmediatamente después de una entrada de capítulo.

```
      \advance\leftskip\@tempdima
      \hskip -\leftskip
      #1\nobreak\hfil
      \nobreak\hb@xt@\@pnumwidth{\hss #2%
        \kern-\p@\kern\p@}\par
    \endgroup
  \fi}
</*article>}
```

En las clases de documento informe y libro, la definición para `\l@section` es mucho más simple.

```
<*report|book>
\newcommand*\l@section{\@dottedtocline{1}{1.5em}{2.3em}}
</*report|book>}
```

\l@subsection

\l@subsubsection

\l@paragraph

\l@subparagraph

\listoffigures

List of figures

Este macro es utilizado para generar la lista de figuras o tablas de figuras.

```
\newcommand\listoffigures{%
<*report|book>
  \if@twocolumn
    \@restonecoltrue\onecolumn
  \else
    \@restonecolfalse
  \fi
  \chapter*{\listfigurename}%
</*report|book>
<*article>  \section*{\listfigurename}%
  \@mkboth{\MakeUppercase\listfigurename}%
          {\MakeUppercase\listfigurename}%
  \@starttoc{lof}%
<report|book>  \if@restonecol\twocolumn\fi
  }
```

\l@figure

El macro `\l@figure` una entrada en la lista de figuras.

```
\newcommand*\l@figure{\@dottedtocline{1}{1.5em}{2.3em}}
```

List of tables

La generación de lista de tablas es de manera similar que la generación de lista de figuras.

\listoftables

Este macro es utilizado para generar la lista de tablas o tablas de tablas.

```
\newcommand\listoftables{%
<*report|book>
  \if@twocolumn
    \@restonecoltrue\onecolumn
  \else
    \@restonecolfalse
  \fi
```

```
  \chapter*{\listtablename}%
</*report|book>
<*article>  \section*{\listtablename}%
  \@mkboth{%
      \MakeUppercase\listtablename}%
      {\MakeUppercase\listtablename}%
  \@starttoc{lot}%
<report|book>  \if@restonecol\twocolumn\fi
  }
```

El macro `\l@table` una entrada en la lista de tablas.

\l @table

```
\let\l@table\l@figure
```

Bibliografía

Cuando está habilitado la opción open de bibliografía, se utiliza la indentación `\bibindent`.

```
\newdimen\bibindent
\setlength\bibindent{1.5em}
```

El entorno `thebibliography` ejecuta los siguientes comandos:

- `\renewcommand{\newblock}{\hskip.11em @plus.33em @minus.07em}`. Este comando redefine el comando `\newblock` qué se utiliza para separar bloques de información en la entrada de bibliografía. Para ver en dónde se utiliza este comando se puede revisar el archivo .bbl.
- `\sloppy`. Este comando hace que el texto de la bibliografía no se salga del margen o es utilizado para realizar saltos de línea en la bibliografía. Las bibliografías pueden contener textos largos y complejos, y a veces es difícil ajustar el texto de manera óptima en las líneas. Al utilizar `\sloppy`, LaTeX se relaja en sus restricciones de ajuste de línea, lo que facilita la creación de saltos de línea adecuados en la bibliografía.
- `\sfcode‘\.=1000\relax`. Este comando afecta el comportamiento del punto (.) en la bibliografía. Normalmente, LaTeX trata un punto como un separador de oraciones y agrega un espacio después de él. Sin embargo, al establecer el `\sfcode‘\.=1000\relax`, se indica que el punto no debe producir un espacio de final de oración en la bibliografía.

La implementación del entorno ’thebibliography’ se basa en el entorno de lista genérico de LaTeX. Utiliza el contador enumiv internamente para generar las etiquetas de la lista, es decir, los números o símbolos que identifican cada entrada de bibliografía.

Es importante tener en cuenta que si se encuentra un entorno vacío de ’thebibliography’, es decir, sin ninguna entrada de bibliografía, se emite una advertencia. Esto sirve como una señal de que podría haber

un error o una omisión en el documento, ya que una bibliografía vacía no es común en un documento académico o científico.

```
\newenvironment{thebibliography}[1]
<*article>
  {\section*{\refname}%
```

El comando @mkboth se ha movido fuera del argumento del encabezado porque, al menos en los estilos report y book (con la opción twocolumn), existen definiciones para `\chapter` que lo eliminarían de lo contrario.

Cuando se utiliza el comando @mkboth en el argumento del encabezado (por ejemplo, en el encabezado de un capítulo), su propósito es establecer los marcadores para el encabezado, que generalmente incluyen el título del capítulo y otra información relevante. Sin embargo, en los estilos report y book con la opción twocolumn, hay definiciones personalizadas para `\chapter` que podrían "tragar" (ignorar o eliminar) el comando @mkboth si se encuentra dentro del argumento del encabezado.

Para evitar que esto suceda y garantizar que @mkboth se ejecute correctamente, se ha movido fuera del argumento del encabezado. De esta manera, el comando @mkboth se ejecuta en un contexto adecuado y no se ve afectado por las definiciones personalizadas de `\chapter` en los estilos mencionados. Esto asegura que los marcadores del encabezado se establezcan correctamente y se reflejen en el documento final.

```
  \@mkboth{\MakeUppercase\refname}{\MakeUppercase\refname}%
</article>
<*!article>
  \chapter*{\bibname}%
  \@mkboth{\MakeUppercase\bibname}{\MakeUppercase\bibname}%
</!article>
\list{\@biblabel{\@arabic\c@enumiv}}%
  {\settowidth\labelwidth{\@biblabel{#1}}%
  \leftmargin\labelwidth
  \advance\leftmargin\labelsep
  \@openbib@code
  \usecounter{enumiv}%
  \let\p@enumiv\@empty
  \renewcommand\theenumiv{\@arabic\c@enumiv}}%
\sloppy
```

Establecemos `\clubpenalty` para todo el entorno de bibliografía.

```
  \clubpenalty4000
  \@clubpenalty \clubpenalty
  \widowpenalty4000%
```

```
    \sfcode`\.\@m}
    {\def\@noitemerr
      {\@latex@warning{Empty `thebibliography' environment}}%
    \endlist}
```

La definición predeterminada de \newblock es producir un pequeño espacio. En este caso, se establece el nuevo valor de \newblock utilizando el comando \hskip para crear un espacio horizontal de .11em con un espacio adicional de .33em y un espacio reducido de .07em.

```
\newcommand\newblock{\hskip .11em\@plus.33em\@minus.07em}
```

La definición predeterminada de @openbib@code es no hacer nada. Este comando será modificado por la opción openbib.

```
\let\@openbib@code\@empty
```

El macro \@biblabel se utiliza para producir la etiqueta de un comando \bibitem[...]. La definición predeterminada utilizada es la del formato LATEX.

```
% \renewcommand*{\@biblabel}[1]{[#1]\hfill}
```

El macro \@cite se utiliza para producir la salida del comando \cite. La definición predeterminada utilizada es la del formato LATEX.

```
% \renewcommand*{\@cite}[1]{[#1]}
```

Index

El entorno 'theindex' se utiliza para crear índices. Este entorno genera un índice con dos columnas, donde cada entrada se trata como un párrafo separado. A nivel de usuario, se utilizan los comandos \item, \subitem y \subsubitem para crear entradas de índice de diferentes niveles.

Cuando se encuentra una nueva letra del alfabeto alfabético en el índice, se puede añadir una cantidad de espacio en blanco utilizando el comando \indexspace. Esto permite agregar un espacio adicional en el índice para separar las entradas de diferentes letras y mejorar la legibilidad.

```
\newenvironment{theindex}
               {\if@twocolumn
                  \@restonecolfalse
                \else
                  \@restonecoltrue
                \fi
<article>                \twocolumn[\section*{\indexname}]%
```

```
<!article>              \twocolumn[\@makeschapterhead{\indexname}]%
                \@mkboth{\MakeUppercase\indexname}%
                        {\MakeUppercase\indexname}%
               \thispagestyle{plain}\parindent\z@
```

Los cambios en los parámetros `\columnseprule` y `\columnsep` deben realizarse después de que `\twocolumn` haya surtido efecto. De lo contrario, pueden afectar la última página antes del índice.

```
               \parskip\z@ \@plus .3\p@\relax
               \columnseprule \z@
               \columnsep 35\p@
               \let\item\@idxitem}
```

Cuando el documento continúa después del índice y era un documento de una sola columna, es necesario volver a una sola columna después del índice.

```
              {\if@restonecol\onecolumn\else\clearpage\fi}
```

Para dar formato a la entrada de índice se definen los siguientes comandos.

```
\newcommand\@idxitem{\par\hangindent 40\p@}
\newcommand\subitem{\@idxitem \hspace*{20\p@}}
\newcommand\subsubitem{\@idxitem \hspace*{30\p@}}
```

La cantidad de espacio en blanco que se inserta entre los "bloques de letras" en el índice.

```
\newcommand\indexspace{\par \vskip 10\p@ \@plus5\p@ \@minus3\p@\relax}
```

Notas al pie

Las notas al pie se separan del cuerpo principal del texto mediante una pequeña línea. Esta línea es dibujada por la macro `\footnoterule`. Es importante asegurarse de que esta línea no ocupe espacio vertical adicional, por lo que compensamos la altura natural de la línea de 0.4pt añadiendo el espacio vertical adecuado. Para evitar que la línea colisione con la nota al pie, primero añadimos un pequeño espacio vertical negativo. Luego colocamos la línea y nos aseguramos de terminar en el mismo punto donde comenzamos esta operación.

```
\renewcommand\footnoterule{%
  \kern-3\p@
  \hrule\@width.4\columnwidth
  \kern2.6\p@}
```

Las enumeraciones de notas al pie se resetean en cada capítulo en clase book y report.

```
<!article> \@addtoreset{footnote}{chapter}
```

El macro `\@makefntext` se utiliza para dar formato a las notas al pie. Este macro recibe como argumento el contenido del nota de pie y utiliza `\@thefnmark` como marca de nota.

```
\newcommand\@makefntext[1]{%
  \@setpar{\@@par
  \@tempdima = \hsize
  \advance\@tempdima-10pt
  \parshape \@ne 10pt \@tempdima}%
  \par
  \parindent 1em\noindent
  \hbox to \z@{\hss\@makefnmark}#1}
```

El efecto de esta definición es que todas las líneas de la nota al pie están sangradas por 10pt, mientras que la primera línea de un nuevo párrafo está sangrada por 1em. Para cambiar estas dimensiones, simplemente sustituye el valor deseado por "10pt" (en ambos lugares) o "1em". La marca está alineada a la derecha junto a la nota al pie.

En estas clases de documentos, utilizamos una macro más sencilla en la cual el texto de la nota al pie se establece como un párrafo de texto normal, sin sangría excepto en la primera línea de un párrafo y en la primera línea de la nota al pie. Por lo tanto, todo lo que la macro debe hacer es establecer `\parindent` con el valor apropiado para los párrafos siguientes y poner la sangría adecuada antes de la marca.

```
\newcommand\@makefntext[1]{%
  \parindent 1em%
  \noindent
  \hb@xt@1.8em{\hss\@makefnmark}#1}
```

La macro @makefnmark se encarga de producir los marcadores de las notas al pie que aparecen en el texto principal. Estos marcadores son los números o símbolos que se colocan junto a las referencias o frases que requieren una nota al pie.

```
\newcommand\@makefnmark{\hbox{\@textsuperscript{\normalfont\@thefnmark}}}
```

16.9 Inicialización

Palabras

```
\newcommand\contentsname{Contents}
\newcommand\listfigurename{List of Figures}
\newcommand\listtablename{List of Tables}
⟨article⟩\newcommand\refname{References}
```

```
⟨report | book⟩\newcommand\bibname{Bibliography}
\newcommand\indexname{Index}
\newcommand\figurename{Figure}
\newcommand\tablename{Table}
\newcommand\partname{Part}
⟨report | book⟩\newcommand\chaptername{Chapter}
\newcommand\appendixname{Appendix}
⟨!book⟩\newcommand\abstractname{Abstract}
```

Fecha

```
\def\today{\ifcase\month\or
  January\or February\or March\or April\or May\or June\or
  July\or August\or September\or October\or November\or December\fi
  \space\number\day, \number\year}
```

Modo dos columnas

```
\setlength\columnsep{10\p@}
\setlength\columnseprule{0\p@}
```

Estilo de página

```
<!book>\pagesytle{plain}
<book>\pagestyle{headings}
\pagenumbering{arabic}
```

Impresión de 2 caras y una sola cara

```
\if@twoside
\else
  \raggedbottom
\fi
\if@twocolumn
  \twocolumn
  \sloppy
  \flushbottom
\else
  \onecolumn
\fi
</article|report|book>
```

CLASES PERSONALIZADAS

17.1 Clase bajo normas APA utilizando TeX y LaTeX

El comportamiento del código de esta plantilla será análogo al código del clase estándar book, por lo que se recomienda conocer el código de una clase book ya explicado.

El manual para la utilización de este plantilla se aloja en el repositorio: `https://github.com/herbermqh/LaTeX-NormasAPA`

```
\NeedsTeXFormat{LaTeX2e}
\ProvidesClass{artapa7}[2023/06/05 artapa7 LaTeX document class]
```

17.2 Módulos

Esta plantilla no se dividirá en módulos debido a que no es tan compleja.

17.3 Código inicial

```
\newcommand\@ptsize{}
\newif\if@restonecol
\newif\if@openright
\newif\if@mainmatter \@mainmattertrue
\newif\if@professional \@professionalfalse
```

17.4 Declaración de opciones

Tamaño de página

```
\if@compatibility\else
  \DeclareOption{a4paper}
     {\setlength\paperheight {297mm}%
      \setlength\paperwidth  {210mm}}
  \DeclareOption{a5paper}
```

```
      {\setlength\paperheight {210mm}%
       \setlength\paperwidth  {148mm}}
   \DeclareOption{b5paper}
      {\setlength\paperheight {250mm}%
       \setlength\paperwidth  {176mm}}
   \DeclareOption{letterpaper}
      {\setlength\paperheight {11in}%
       \setlength\paperwidth  {8.5in}}
   \DeclareOption{legalpaper}
      {\setlength\paperheight {14in}%
       \setlength\paperwidth  {8.5in}}
   \DeclareOption{executivepaper}
      {\setlength\paperheight {10.5in}%
       \setlength\paperwidth  {7.25in}}
   \DeclareOption{landscape}
      {\setlength\@tempdima   {\paperheight}%
       \setlength\paperheight {\paperwidth}%
       \setlength\paperwidth  {\@tempdima}}
\fi
\newlength\margin
\setlength\margin{1in}
```

Tamaño de letra

```
\if@compatibility
  \renewcommand\@ptsize{0}
\else
  \DeclareOption{10pt}{\renewcommand\@ptsize{0}}
\fi
\DeclareOption{11pt}{\renewcommand\@ptsize{1}}
\DeclareOption{12pt}{\renewcommand\@ptsize{2}}
```

Impresión de una cara o de 2 caras

```
\if@compatibility\else
  \DeclareOption{oneside}{\@twosidefalse \@mparswitchfalse}
\fi
\DeclareOption{twoside}{\@twosidetrue  \@mparswitchtrue}
```

Opciones de borrador

```
\DeclareOption{draft}{\setlength\overfullrule{5pt}}
\if@compatibility\else
  \DeclareOption{final}{\setlength\overfullrule{0pt}}
\fi
```

Openright option

```
\if@compatibility
\@openrighttrue
\else
\DeclareOption{openright}{\@openrighttrue}
\DeclareOption{openany}{\@openrightfalse}
\fi
```

Impresión de 2 columnas

```
\if@compatibility\else
  \DeclareOption{onecolumn}{\@twocolumnfalse}
\fi
\DeclareOption{twocolumn}{\@twocolumntrue}
```

Numeración de ecuaciones a la izquierda

```
\DeclareOption{leqno}{\input{leqno.clo}}
```

Flush left displays

```
\DeclareOption{fleqn}{\input{fleqn.clo}}
```

Open bibliography

```
\DeclareOption{openbib}{%
  \AtEndOfPackage{%
    \renewcommand\@openbib@code{%
      \advance\leftmargin\bibindent
      \itemindent -\bibindent
      \listparindent \itemindent
      \parsep \z@
      }%
```

```
    \renewcommand\newblock{\par}}%
}
```

17.5 Ejecución de opciones

```
\DeclareOption{professional}{\@professionaltrue}
\ExecuteOptions{letterpaper,10pt,oneside,onecolumn,final,openright}
\ProcessOptions
```

17.6 Paquetes utilizado o cargados

```
\RequirePackage[utf8]{inputenc}
\RequirePackage[T1]{fontenc}
```

17.7 Diseño del documento

Fuentes

```
\input{bk1\@ptsize.clo}
\usepackage{microtype}
```

Párrafos

```
\setlength\lineskip{0\p@}
\setlength\normallineskip{0\p@}
\renewcommand\baselinestretch{2}
\setlength\parskip{0\p@}
\setlength\parindent{1.27cm}
\@lowpenalty   51
\@medpenalty  151
\@highpenalty 301
```

Page layout

Separación vertical

```
\setlength\textheight{\dimexpr\paperheight-\margin*2}
\setlength\headheight{\baselineskip}
\setlength\headsep   {\dimexpr.5\margin-.5\headheight}
\setlength\topmargin{\dimexpr-\headheight-\headsep}
% \setlength\topskip{1\p@}
```

Dimensión del texto

```
\setlength\footskip{\dimexpr.5\margin+0.5\baselineskip\relax}
\setlength\textwidth{\dimexpr\paperwidth-\margin*2}
```

Márgenes

```
\setlength\marginparsep{0\p@}
\setlength\oddsidemargin{0\p@}
\setlength\evensidemargin{0\p@}
\setlength\marginparwidth{0\p@}
```

Notas en pie de página

```
% \setlength\footnotesep{0\p@}
% \setlength{\skip\footins}{9\p@ \@plus 4\p@ \@minus 2\p@}
```

Float placement parameters

```
\setcounter{topnumber}{2}
\renewcommand\topfraction{.7}
\setcounter{bottomnumber}{1}
\renewcommand\bottomfraction{.3}
\setcounter{totalnumber}{3}
\renewcommand\textfraction{.2}
\renewcommand\floatpagefraction{.5}
\setcounter{dbltopnumber}{2}
\renewcommand\dbltopfraction{.7}
\renewcommand\dblfloatpagefraction{.5}
```

Estilo de página

Convención de marcado

Lleva las mismas convenciones de marcado que una clase book.

Definiendo estilo de página

```
\if@twoside
  \def\ps@headings{%
      \let\@oddfoot\@empty\let\@evenfoot\@empty
```

```
      \def\@evenhead{\thepage\hfil\if@professional\ifdef\@shorttitle{\MakeUp
      \def\@oddhead{\if@professional\ifdef\@shorttitle{\MakeUppercase\@short
      \let\@mkboth\markboth}
\else
  \def\ps@headings{%
    \let\@oddfoot\@empty
    \def\@oddhead{\if@professional\ifdef\@shorttitle{\MakeUppercase\@shortti
    \let\@mkboth\markboth}
\fi
\def\ps@myheadings{%
    \let\@oddfoot\@empty\let\@evenfoot\@empty
    \def\@evenhead{\thepage\hfil\slshape\leftmark}%
    \def\@oddhead{{\slshape\rightmark}\hfil\thepage}%
    \let\@mkboth\@gobbletwo
    \let\chaptermark\@gobble
    \let\sectionmark\@gobble}
```

17.8 Marcado de un documento

El título

```
\DeclareRobustCommand*{\shorttitle}[1]{\gdef\@shorttitle{#1}}
\DeclareRobustCommand*{\affiliation}[1]{\gdef\@affiliation{#1}}
\DeclareRobustCommand*{\course}[1]{\gdef\@course{#1}}
\DeclareRobustCommand*{\professor}[1]{\gdef\@professor{#1}}
\DeclareRobustCommand*{\authornote}[1]{\gdef\@authornote{#1}}
```

```
\if@professional
  \newcommand\maketitle{\begin{titlepage}
  \null
  \vskip4\baselineskip
  \begin{center}%
    {\LARGE\bfseries\@title\par}%
    \vskip 2\baselineskip%
    {\large \@author}
    \vskip \baselineskip%
    {\large \@affiliation}
  \end{center}
  \vfil
  \centerline{\Large \bfseries Notas del autor}
```

```
  \begin{flushleft}
    \@authornote
  \end{flushleft}
  \vfil\null
  \end{titlepage}%
}
\else
  \newcommand\maketitle{\par
    \begingroup
      \if@twocolumn
        \ifnum \col@number=\@ne
          \@maketitle
        \else
          \twocolumn[\@maketitle]%
        \fi
      \else
      \newpage
        \global\@topnum\z@
        \@maketitle
      \fi
      \thispagestyle{headings}
    \endgroup
  }
```

```
\def\@maketitle{%
  \newpage
  \null
  \vskip 4\baselineskip%
  \begin{center}%
    {\LARGE\bfseries\@title\par}%
    \vskip 2\baselineskip%
    {\large \@author}
    \vskip \baselineskip%
    {\large \@affiliation}
    \vskip \baselineskip%
    {\large \@course}
    \vskip \baselineskip%
    {\large \@professor}
    \vskip \baselineskip%
    {\large \@date}%
  \end{center}}
\fi
```

Capítulos y secciones

Definir contadores

```
\setcounter{secnumdepth}{4} %determina el nivel de profundidad de los números de sección enumeradas a ingresar a toc
\newcounter {part}
\newcounter {chapter}
\newcounter {section}[chapter]
\newcounter {subsection}[section]
\newcounter {subsubsection}[subsection]
\newcounter {paragraph}[subsubsection]
\renewcommand \thepart {\@Roman\c@part}
\renewcommand \thechapter {\@arabic\c@chapter}
\renewcommand \thesection {\thechapter.\@arabic\c@section}
\renewcommand\thesubsection   {\thesection.\@arabic\c@subsection}
\renewcommand\thesubsubsection{\thesubsection.\@arabic\c@subsubsection}
\renewcommand\theparagraph    {\thesubsubsection.\@arabic\c@paragraph}
\newcommand\@chapapp{\chaptername}
```

Frontmatter, mainmatter y backmatter

```
\newcommand\frontmatter{%
  \cleardoublepage
  \@mainmatterfalse
  \pagenumbering{arabic}}
\newcommand\mainmatter{%
  \cleardoublepage
  \@mainmattertrue
  \pagenumbering{arabic}}
\newcommand\backmatter{%
  \if@openright
    \cleardoublepage
  \else
    \clearpage
  \fi
  \@mainmatterfalse}
```

Partes

Capítulos

```
\newcommand\chapter{\global\@topnum\z@
                    \secdef\@chapter\@schapter}
\def\@chapter[#1]#2{\ifnum \c@secnumdepth >\m@ne
                       \if@mainmatter
                         \refstepcounter{chapter}%
                         \addcontentsline{toc}{chapter}%
                                    {\protect\numberline{\thechapter}#1}%
                       \else
                         \if@openright\cleardoublepage\else\clearpage\fi
                         \addcontentsline{toc}{chapter}{#1}%
                       \fi
                    \else
                      \addcontentsline{toc}{chapter}{#1}%
                    \fi
                    \addtocontents{lof}{\protect\addvspace{10\p@}}%
                    \addtocontents{lot}{\protect\addvspace{10\p@}}%
                    \if@twocolumn
                      \clearpage
                      \@topnewpage[\@makechapterhead{#2}]%
                    \else
                      \par\@makechapterhead{#2}%
                      \@afterheading
                    \fi}
\def\@schapter#1{%
                \if@openright\cleardoublepage\else\clearpage\fi
                \if@twocolumn
                  \clearpage
                  \@topnewpage[\@makechapterhead{#1}]%
                \else
                  \@makechapterhead{#1}%
                  \@afterheading
                \fi}
\def\@makechapterhead#1{%
  {\parindent \z@ \centering \normalfont
  \interlinepenalty\@M
  \bfseries #1\par\nobreak
  }}
```

Encabezados de nivel inferior

```
\def\@section[#1]#2{%
  \ifnum1>\c@secnumdepth
    \addcontentsline{toc}{section}{#1}%
  \else
    \refstepcounter{section}%
    \addcontentsline{toc}{section}{\protect\numberline{\thesection}#2}%
  \fi
  \@makesectionhead{#2}}
\def\@ssection#1{\begingroup\@makesectionhead{#1}\endgroup}
\def\@makesectionhead#1{%
  {\par\parindent \z@ \raggedright \normalfont
  \interlinepenalty\@M
  \bfseries #1\par\nobreak
  }}
\newcommand\subsection{\secdef\@subsection\@ssubsection}
\def\@subsection[#1]#2{%
  \ifnum2>\c@secnumdepth
    \addcontentsline{toc}{subsection}{#1}%
  \else
    \refstepcounter{subsection}%
    \addcontentsline{toc}{subsection}{\protect\numberline{\thesubsection}#2}
  \fi
  \@makesubsectionhead{#2}}
\def\@ssubsection#1{\begingroup\@makesubsectionhead{#1}\endgroup}
\def\@makesubsectionhead#1{%
  {\par\parindent \z@ \raggedright \normalfont
  \interlinepenalty\@M
  \bfseries\textit{#1}\par\nobreak
  }}
\newcommand\subsubsection{\secdef\@subsubsection\@ssubsubsection}
\def\@subsubsection[#1]#2{%
  \ifnum3>\c@secnumdepth
    \addcontentsline{toc}{subsubsection}{#1}%
  \else
    \refstepcounter{subsubsection}%
    \addcontentsline{toc}{subsubsection}{\protect\numberline{\thesubsubsecti
  \fi
  \@makesubsubsectionhead{#2}}
```

```
\def\@ssubsubsection#1{\begingroup\@makesubsubsectionhead{#1}\endgroup}
\def\@makesubsubsectionhead#1{%
  {\par\normalfont
  \interlinepenalty\@M
  \bfseries#1.\nobreak
  }}
\newcommand\paragraph{\secdef\@paragraph\@sparagraph}
\def\@paragraph[#1]#2{%
  \ifnum4>\c@secnumdepth
    \addcontentsline{toc}{paragraph}{#1}%
  \else
    \refstepcounter{paragraph}%
    \addcontentsline{toc}{paragraph}{#2}%
  \fi
  \@makeparagraphhead{#2}}
\def\@sparagraph#1{\begingroup\@makeparagraphhead{#1}\endgroup}
\def\@makeparagraphhead#1{%
  {\par\normalfont
  \interlinepenalty\@M
  \bfseries\textit{#1.}\nobreak
  }}
```

Listas

Parámetros generales de una lista

```
\if@twocolumn
  \setlength\leftmargini  {2em}
\else
  \setlength\leftmargini  {2.5em}
\fi
\leftmargin  \leftmargini
\setlength\leftmarginii  {2.2em}
\setlength\leftmarginiii {1.87em}
\setlength\leftmarginiv  {1.7em}
\if@twocolumn
  \setlength\leftmarginv  {.5em}
  \setlength\leftmarginvi {.5em}
\else
  \setlength\leftmarginv  {1em}
  \setlength\leftmarginvi {1em}
\fi
```

```
\setlength  \labelsep  {.5em}
\setlength  \labelwidth{\leftmargini}
\addtolength\labelwidth{-\labelsep}
\@beginparpenalty -\@lowpenalty
\@endparpenalty   -\@lowpenalty
\@itempenalty     -\@lowpenalty
```

Enumerate

```
\renewcommand\theenumi{\@arabic\c@enumi}
\renewcommand\theenumii{\@alph\c@enumii}
\renewcommand\theenumiii{\@roman\c@enumiii}
\renewcommand\theenumiv{\@Alph\c@enumiv}
\newcommand\labelenumi{\theenumi.}
\newcommand\labelenumii{(\theenumii)}
\newcommand\labelenumiii{\theenumiii.}
\newcommand\labelenumiv{\theenumiv.}
\renewcommand\p@enumii{\theenumi}
\renewcommand\p@enumiii{\theenumi(\theenumii)}
\renewcommand\p@enumiv{\p@enumiii\theenumiii}
```

Itemize

```
\newcommand\labelitemi  {\labelitemfont \textbullet}
\newcommand\labelitemii {\labelitemfont \bfseries \textendash}
\newcommand\labelitemiii{\labelitemfont \textasteriskcentered}
\newcommand\labelitemiv{ \labelitemfont \textperiodcentered}
\newcommand\labelitemfont{\normalfont}
```

Description

```
\newenvironment{description}
               {\list{}{\labelwidth\z@ \itemindent-\leftmargin
                        \let\makelabel\descriptionlabel}}
               {\endlist}
\newcommand*\descriptionlabel[1]{\hspace\labelsep
                                \normalfont\bfseries #1}
```

Definición de nuevos entornos

```
\newenvironment{verse}
               {\let\\\@centercr
```

```
                \list{}{\itemsep      \z@
                        \itemindent   -1.5em%
                        \listparindent\itemindent
                        \rightmargin  \leftmargin
                        \advance\leftmargin 1.5em}%
                \item\relax}
               {\endlist}
\newenvironment{quotation}
               {\list{}{\listparindent 1.5em%
                        \itemindent    \listparindent
                        \rightmargin   \leftmargin
                        \parsep        \z@ \@plus\p@}%
                \item\relax}
               {\endlist}
\newenvironment{quote}
               {\list{}{\leftmargin1.27cm\rightmargin0pt}%
                \item\relax}
               {\endlist}
```

Titlepage

```
\if@compatibility
  \newenvironment{titlepage}
    {%
      \cleardoublepage
      \if@twocolumn
        \@restonecoltrue\onecolumn
      \else
        \@restonecolfalse\newpage
      \fi
      \thispagestyle{headings}
      \setcounter{page}\z@
    }%
    {\if@restonecol\twocolumn \else \newpage \fi
    }
\else
  \newenvironment{titlepage}
    {%
      \cleardoublepage
      \if@twocolumn
        \@restonecoltrue\onecolumn
      \else
```

```
        \@restonecolfalse\newpage
      \fi
      \thispagestyle{headings}
      \setcounter{page}\@ne
    }%
    {\if@restonecol\twocolumn \else \newpage \fi
     \if@twoside\else
        \setcounter{page}\@ne
     \fi
    }
\fi
```

Appendix

```
\newcommand\appendix{\par
  \setcounter{chapter}{0}%
  \setcounter{section}{0}%
  \gdef\@chapapp{\appendixname}%
  \gdef\thechapter{\@Alph\c@chapter}}
```

Configuración de parámetros para entornos existentes

Matrix and tabular

```
\setlength\arraycolsep{5\p@}
\setlength\tabcolsep{6\p@}
\setlength\arrayrulewidth{.4\p@}
\setlength\doublerulesep{2\p@}
```

Tabbing

```
\setlength\tabbingsep{\labelsep}
\skip\@mpfootins = \skip\footins
\setlength\fboxsep{3\p@}
\setlength\fboxrule{.4\p@}
```

Minipage

Equation and eqnarray

```
\@addtoreset {equation}{chapter}
\renewcommand\theequation
  {\ifnum \c@chapter>\z@ \thechapter.\fi \@arabic\c@equation}
```

Objetos flotantes

Figure

```
\newcounter{figure}
\renewcommand \thefigure{\@arabic\c@figure}
\def\fps@figure{tbp}
\def\ftype@figure{1}
\def\ext@figure{lof}
\def\fnum@figure{\figurename\nobreakspace\thefigure}
\newenvironment{figure}
               {\@float{figure}}
               {\end@float}
\newenvironment{figure*}
               {\@dblfloat{figure}}
               {\end@dblfloat}
```

Table

```
\newcounter{table}
\renewcommand \thetable{\@arabic\c@table}
\def\fps@table{tbp}
\def\ftype@table{2}
\def\ext@table{lot}
\def\fnum@table{\tablename\nobreakspace\thetable}
\newenvironment{table}
               {\@float{table}}
               {\end@float}
\newenvironment{table*}
               {\@dblfloat{table}}
               {\end@dblfloat}
```

Subtítulos o caption

```
\newlength\abovecaptionskip
\newlength\belowcaptionskip
```

```
\setlength\abovecaptionskip{\baselineskip}
\setlength\belowcaptionskip{\baselineskip}
\long\def\@makecaption#1#2{%
  \vskip\abovecaptionskip
  {\bfseries #1} \newline \textit{#2}
  \vskip\belowcaptionskip}
```

Cambio de fuente

```
\DeclareOldFontCommand{\rm}{\normalfont\rmfamily}{\mathrm}
\DeclareOldFontCommand{\sf}{\normalfont\sffamily}{\mathsf}
\DeclareOldFontCommand{\tt}{\normalfont\ttfamily}{\mathtt}
\DeclareOldFontCommand{\bf}{\normalfont\bfseries}{\mathbf}
\DeclareOldFontCommand{\it}{\normalfont\itshape}{\mathit}
\DeclareOldFontCommand{\sl}{\normalfont\slshape}{\@nomath\sl}
\DeclareOldFontCommand{\sc}{\normalfont\scshape}{\@nomath\sc}
\DeclareRobustCommand*\cal{\@fontswitch\relax\mathcal}
\DeclareRobustCommand*\mit{\@fontswitch\relax\mathnormal}
\newcommand\@pnumwidth{1.55em}
\newcommand\@tocrmarg{2.55em}
\newcommand\@dotsep{4.5}
```

17.9 Referencias cruzadas

Tabla de contenido y otros

Table of contents

\tableofcontents

```
\setcounter{tocdepth}{4}
```

```
\newcommand\tableofcontents{%
    \if@twocolumn
      \@restonecoltrue\onecolumn
    \else
      \@restonecolfalse
    \fi
    \chapter*{\contentsname}%
    \addcontentsline{toc}{chapter}{\contentsname}%
    \@starttoc{toc}%
    \if@restonecol\twocolumn\fi}
\newcommand*\l@chapter[2]{%
  \ifnum \c@tocdepth >\m@ne
    \addpenalty{-\@highpenalty}%
```

```
        \vskip 1.0em \@plus\p@
        \setlength\@tempdima{1.5em}%
        \begingroup
          \parindent \z@ \rightskip \@pnumwidth
          \parfillskip -\@pnumwidth
          \leavevmode \bfseries
          \advance\leftskip\@tempdima
          \hskip -\leftskip
          #1\nobreak\hfil
          \nobreak\hb@xt@\@pnumwidth{\hss #2%
                                     \kern-\p@\kern\p@}\par
          \penalty\@highpenalty
        \endgroup
      \fi}
    \newcommand*\l@section{\@dottedtocline{1}{1.5em}{2.3em}}
    \newcommand*\l@subsection{\@dottedtocline{2}{3.8em}{3.2em}}
    \newcommand*\l@subsubsection{\@dottedtocline{3}{7.0em}{4.1em}}
    \newcommand*\l@paragraph{\@dottedtocline{4}{10em}{5em}}
```

List of figures

```
    \newcommand\listoffigures{%
        \if@twocolumn
          \@restonecoltrue\onecolumn
        \else
          \@restonecolfalse
        \fi
        \chapter*{\listfigurename}%
        \addcontentsline{toc}{chapter}{\listfigurename}%
        \@starttoc{lof}%
        \if@restonecol\twocolumn\fi
        }
    \newcommand*\l@figure{\@dottedtocline{1}{1.5em}{2.3em}}
```

List of tables

```
    \newcommand\listoftables{%
        \if@twocolumn
          \@restonecoltrue\onecolumn
        \else
          \@restonecolfalse
        \fi
```

```
    \chapter*{\listtablename}%
    \addcontentsline{toc}{chapter}{\listtablename}%
    \@starttoc{lot}%
    \if@restonecol\twocolumn\fi
    }
```

```
\let\l@table\l@figure
```

Biografía

```
\newdimen\bibindent
\setlength\bibindent{1.5em}
\newenvironment{thebibliography}[1]
     {\chapter*{\bibname}%
      \@mkboth{\MakeUppercase\bibname}{\MakeUppercase\bibname}%
      \list{\@biblabel{\@arabic\c@enumiv}}%
           {\settowidth\labelwidth{\@biblabel{#1}}%
            \leftmargin\labelwidth
            \advance\leftmargin\labelsep
            \@openbib@code
            \usecounter{enumiv}%
            \let\p@enumiv\@empty
            \renewcommand\theenumiv{\@arabic\c@enumiv}}%
      \sloppy
      \clubpenalty4000
      \@clubpenalty \clubpenalty
      \widowpenalty4000%
      \sfcode`\.\@m}
     {\def\@noitemerr
       {\@latex@warning{Empty `thebibliography' environment}}%
      \endlist}
\newcommand\newblock{\hskip .11em\@plus.33em\@minus.07em}
\let\@openbib@code\@empty
```

Citas

```
\long\def\pcite{\@ifnextchar[ \@cite@parentetico@textual\@cite@parentetico@parafraseado}
\long\def\@cite@parentetico@parafraseado#1#2{#2 \citep{#1}\everypar{}}
```

```
\long\def\@cite@parentetico@textual[#1] #2#3{\everypar{\hangindent\parindent}#3 \citep{#2}\everypar{}}
\long\def\ncite{\@ifnextchar[ \@cite@narrativo@textual\@cite@narrativo@parafraseado}
\long\def\@cite@narrativo@parafraseado#1#2{\citet{#1} #2}
\long\def\@cite@narrativo@textual[#1] #2#3{\hangindent1.27cm\everypar{\hangindent1.27cm}\citet{#2} #3 (#1)\everypar{}}
```

Index

```
\newenvironment{theindex}
               {\if@twocolumn
                  \@restonecolfalse
                \else
                  \@restonecoltrue
                \fi
                \twocolumn[\@makeschapterhead{\indexname}]%
                \@mkboth{\MakeUppercase\indexname}%
                        {\MakeUppercase\indexname}%
                \thispagestyle{plain}\parindent\z@
                \parskip\z@ \@plus .3\p@\relax
                \columnseprule \z@
                \columnsep 35\p@
                \let\item\@idxitem}
               {\if@restonecol\onecolumn\else\clearpage\fi}
\newcommand\@idxitem{\par\hangindent 40\p@}
\newcommand\subitem{\@idxitem \hspace*{20\p@}}
\newcommand\subsubitem{\@idxitem \hspace*{30\p@}}
\newcommand\indexspace{\par \vskip 10\p@ \@plus5\p@ \@minus3\p@\relax}
```

Notas al pie

```
\renewcommand\footnoterule{%
  \kern-3\p@
  \hrule\@width.4\columnwidth
  \kern2.6\p@}
\@addtoreset{footnote}{chapter}
\newcommand\@makefntext[1]{%
    \parindent 1em%
```

```
    \noindent
    \hb@xt@1.8em{\hss\@makefnmark}#1}
```

17.10 Inicialización

Palabras

```
\newcommand\contentsname{Contenido}
\newcommand\listfigurename{Lista de figura}
\newcommand\listtablename{Lista de tablas}
\newcommand\bibname{Bibliograf\'ia}
\newcommand\indexname{Índice}
\newcommand\figurename{Figura}
\newcommand\tablename{Tabla}
\newcommand\partname{Parte}
\newcommand\chaptername{Capítulo}
\newcommand\appendixname{Apendice}
```

Otros definiciones

```
\usepackage{natbib}
\renewcommand\cite{\begingroup\NAT@fullfalse\let\NAT@ctype\z@\NAT@partrue
    \@ifstar{\NAT@swatrue\NAT@citetp}{\NAT@swafalse\NAT@citetp}}
```

```
\long\def\note#1{\vskip\baselineskip\textit{Nota.} #1\hfill}
```

Fecha

```
\def\today{\ifcase\month\or
  January\or February\or March\or April\or May\or June\or
  July\or August\or September\or October\or November\or December\fi
  \space\number\day, \number\year}
```

Modo dos columnas

```
\setlength\columnsep{10\p@}
\setlength\columnseprule{0\p@}
```

Estilo de página

```
\pagestyle{headings}
\pagenumbering{arabic}
```

Impresión de 2 caras y una sola cara

```
\if@twoside
\else
  \raggedbottom
\fi
\if@twocolumn
  \twocolumn
  \sloppy
  \flushbottom
\else
  \onecolumn
\fi
\endinput
```

17.11 Clase para libros utilizando TeX y LaTeX

Además de crear un documentclass como artapa7 también podemos crear otras plantillas para libros sin utilizar paquetes como titlesec.

Bibliografía

[A. and F.,] A., A. B. and F., W. M. Edición de textos científicos latex.

[Abrahams et al., 1990] Abrahams, P. W., Hargreaves, K. A., and Berry, K. (1990). TEX for the impatient. Addison-Wesley, Reading, Mass., reprinted with corrections edition.

[Goossens, 2010] Goossens, M., editor (2010). The LATEX graphics companion. Addison-Wesley series on tools and techniques for computer typesetting. Addison-Wesley, Upper Saddle River, NJ [u.a.], 2. ed., 3. printing edition. Box u.d.T.: The LATEX companions. Rev. set.

[Grätzer, 1996] Grätzer, G. (1996). Math into LATEX. Birkhäuser, Boston [u.a.]. Literaturverz. S. [413] - 415.

[Grätzer, 2016] Grätzer, G. (2016). More math into latex.

[Jackson and Goossens,] Jackson, S. and Goossens, L. Handbook of adolescent development.

[Knuth, 2000] Knuth, D. E. (2000). Computers & typesetting. Volume A–E. Boxed set. Volume A: The TeXbook. Volume B: TeX: the program. Volume C: The Metafontbook. Volume D: Metafont: the program. Volume E: Computer modern typefaces. Amsterdam : Addison-Wesley Longman, updated Millenium ed. edition.

[Kottwitz, 2011] Kottwitz, S. (2011). Latex beginner's guide. Includes index.

[Lamport, 1995] Lamport, L. (1995). Das LATEX-Handbuch. Addison-Wesley, Bonn [u.a.], [2. aufl.] edition. Literaturverz. S. [279].

[Lamport, 1998] Lamport, L. (1998). LATEX. Addison-Wesley, Reading, Mass. [u.a.], 2nd ed., 7. + 8. printing edition. Bibliography S. 235.

[Lamport et al., 2022] Lamport, L., Mittelbach, F., and Braams, J. (2022). Standard document classes for latex version 2e. pages 1–58.

[Man, 1992] Man, K. (1992). Tex for the impatient. 14:238.

[Mittelbach and Goossens, 2005] Mittelbach, F. and Goossens, M. (2005). Der LATEX-Begleiter. st - scientific tools. Pearson Studium, München [u.a.], 2., überarb. und erw. aufl. edition. Hier auch später erschienene, unveränderte Nachdrucke.

[Mittelbach et al., 2013] Mittelbach, F., Goossens, M., and Braams, J., editors (2013). The LATEX companion. Addison-Wesley series on tools and techniques for computer typesetting. Addison-Wesley, Boston, Mass. [u.a.], 2. ed., 12. print. edition. Literaturverz. S. [963] - 982.

[Mojžišová and Pócsová, 2018] Mojžišová, A. and Pócsová, J. (2018). Visualisation of mathematical content using latex animations. pages 536–541, Szilvasvarad, Hungary. IEEE.

[Spivak, 1990] Spivak, M. D. (1990). The joy of TEX. American Mathematical Society, Providence, RI, 2. ed edition.

[Syropoulos, 2003] Syropoulos, A. (2003). Digital typography using latex. Includes bibliographical references (p. 469-470) and indexes.

[Tantau, 2007] Tantau, T. (2007). Tutoriel tikz. pages 23–92.

[van Dongen, 2012] van Dongen, M. R. C. (2012). Latex and friends.

Lista de figuras